

# SYM 1541 DOS

Robert Jordan, 1984

## **Contents:**

1. SYM 1541 DOS Manual
2. Hardware interface circuit Design
3. Source of 1541 DOS, RAE Basic Monitor driver
4. Crossreference of software
5. Contents of Utility Disk

**SYM-1 DISK OPERATING SYSTEM FOR THE COMMODORE  
1541 DISK DRIVE**

**MONITOR LINKS**

**RAE LINKS**

**BAS LINKS**

**COPYRIGHT (C) 1984 by Ronald A. Jordan**

**Distributed by**

**JORDAN & ASSOCIATES**

**2611 Madrono Drive**

**Ann Arbor, MI 48103**



# SYM-1 1541 DOS SYSTEM

## TABLE OF CONTENTS

INTRODUCTION . . . . .	3
INSTALLATION AND SET UP . . . . .	5
SPECIAL INSTRUCTIONS FOR THE EPROM OPTION .	6
COMMAND SUMMARY . . . . .	7
USING THE SYSTEM . . . . .	8
A. USING THE MONITOR LINK . . . . .	8
B. USING THE RAE LINK . . . . .	9
C. USING THE BASIC LINK . . . . .	10
ERROR CODES . . . . .	11
TABLE OF SYSTEM RAM VECTORS . . . . .	12
TABLE OF SYSTEM AND PRIMITIVE VECTORS . . .	13
UTILITIES AND ENHANCEMENTS, DISCLAIMER, AND COPYRIGHT NOTICE . . . . .	14
PRICES AND ORDERING INFORMATION . . . . .	15

## SYM-1 1541 DOS SYSTEM

### INTRODUCTION

The SYM-1 DOS for the Commodore 1541 disk drive greatly expands the capability and compatibility of the SYM-1. Although several disk systems are available for the SYM-1, all are relatively expensive. In addition, each offers its own unique disk formatting, which prevents disk interchangeability and greatly limits access to commercial and public software. The SYM-1 1541 DOS helps to fill this gap by using the Commodore 1541 disk drive to create Commodore compatible disks. Since the Commodore 1541 has the DOS built into it, the SYM-1 DOS can take advantage of the Commodore DOS features and reside in RAM or EPROM very compactly (approx. 3K). It can function equally well in an unexpanded SYM-1, a development system, or even a fully expanded system. With the installation of SYM-1 1541 DOS the SYM-1 can become a much more powerful little computer that is easier and more enjoyable to use.

Functionally, the SYM-1 1541 DOS consists of four modules: the Primitive routines, the Monitor link, the RAE link, and the BAS link. The primitives include all of the low level routines needed to communicate with the Commodore 1541 disk drive over the serial bus. The SYM-1 has several different VIA ports that could be connected to the serial bus. However, the primitive interface routines are dependent on the selected bus configuration on the VIA. The standard VIA port configuration uses VIA #1 (Port A) on the A-connector. Other configurations are available upon request at a nominal fee. The Monitor link interfaces with SUPERMON. All commands are vectored through the unrecognized syntax vector (URSVEC) and may be easily enhanced or altered as desired. When system vectors are used, a return vector is placed in system RAM for additional patches. The commands include load and save memory to disk with the option for a relocated load. Other commands allow easy display of the disk directory, reading the error channel, changing the device number to another drive, and sending Commodore 1541 DOS commands. The assembler editor (RAE) link includes the monitor disk commands which are implemented through the DC command. The load and save commands use special forms of the PUT and GET commands. The load command will load RAE source files with the option for an append and the save will save the source

## SYM-1 1541 DOS SYSTEM

files. Files may also be assembled from disk. To enter RAE, a simple monitor jump command is used which then completely configures the file parameters for a 28K (or whatever size desired) system. The monitor may be reentered with a control C and all of the monitor commands are still available. To start BASIC a simple monitor jump command is also used, which configures BASIC for a 28K system with 80 columns and then patches in the new command processor using INVEC and OUTVEC. The disk commands are implemented through OUTVEC so that future commands maybe added easily and used under program control. Currently, commands to load and save program files to disk are supported. The other disk commands are also available. It is possible to exit BASIC with a control C and then warm start it again with a .G command without the loss of the BASIC text. Normal cassette I/O is functional in BASIC, RAE, and SUPERMON.

The SYM-1 1541 DOS system includes the following:

1. Hardware interface module for the serial bus connection to the SYM-1. VIA #1, Port A. (optional configurations available)
2. Complete source listing for the primitives, monitor, RAE, and BAS links with Cross Referenced Label Listing
3. Cassette tape with object code.  
(normal start address \$7000, but others available at no charge)
4. SYM-1 1541 DOS manual.
5. EPROM with object code for primitives and monitor links. (optional)
6. Source files on disk or cassette.  
(optional)

## SYM-1 1541 DOS SYSTEM

### INSTALLATION AND SET UP

The SYM-1 1541 Disk Operating System is easy to set up and run. The interface module supplied connects onto the A-connector of the SYM-1 board so that the side of the 22/44 connector with the most wires is on the top side of the SYM-1 board. Please make sure that all power is off to the SYM-1 and the disk drive before making any connections. If a special I/O configuration was ordered, please see the special instructions for the installation. The microclip must be connected to the reset signal of the SYM-1 board located at connector pins AA-13 or E-7 (res). This allows the disk drive to be reset when the SYM-1 is reset. The disk drive din plug is connected into the interface module jack. If additional 1541 drives are used, they should be chained using the extra connector on the back of each drive according to the instructions in the drive manual. The DOS system software assumes the main drive to be device #8. A second drive can be used as device #9. It is possible to connect and address more than two 1541 disk drives, but the device number must be set up through the Monitor Link command (SC #x) using 0A hex (#10) or 0B hex (#11). Device numbers less than 8 are not supported except in the RAE Link where device #0 is the cassette tape. To install the EPROM, if ordered, please see the special instructions.

## SYM-1 1541 DOS SYSTEM

### SPECIAL INSTRUCTIONS FOR THE EPROM OPTION

If you ordered the EPROM option you received a 2716 (2K) EPROM containing the Monitor Link and the Primitive Routines. The cassette tape supplied contains the object code for the RAE and BAS Links plus several other relocated versions of the entire system object code, which should be helpful in customizing your system. To install the EPROM, select the appropriate ROM socket on the SYM-1 board and set the jumpers for the desired socket, for the correct start address shown on the EPROM label, and for the 2716 type EPROM. Please be sure that all power is off when making these changes. Insert the EPROM into the socket carefully, remembering that it is static sensitive. After connecting the interface module as described in the section on Installation and Set Up, the System is ready to use.

The standard version of the SYM-1 1541 DOS is located at \$7000-\$7C75, which corresponds with the source code listing in the manual. The EPROM version assumes that the RAE and BAS Links are located at \$7800-\$7C75. If you find it necessary to use the RAE and BAS Link portions of another relocated version of the object code, the high addresses of the J 0 (BASIC cold start vector) and J 4, J 5 (RAE warm and cold start vectors) must be changed, in addition to several jumps to system routines. To consolidate the DOS routines, you might consider installing a 2K static RAM chip, such as a 6116 in the 2K memory space above your EPROM for the RAE and BAS Links. The RAE and BAS Links were not put into EPROM so that both may be expanded and enhanced. As each portion increases in size, they probably will no longer reside concurrently in the limited 2K memory space. However, it will be possible to overlay the Links in memory as necessary when running RAE or BASIC.

## SYM-1 1541 DOS SYSTEM

### COMMAND SUMMARY

#### MONITOR LINK:

1. S2 xxxx,yyyy/FILENAME  
save memory to disk with the name
2. L2 /FILENAME load memory  
L2 xxxx/FILENAME relocated memory load
3. CONTROL D reenter file name
4. SC #x change device number  
SC ! read error channel  
SC ? list directory  
BREAK Key, pause listing  
SPACE BAR, continue listing  
SC /DISKCOMMAND send disk command
5. J0 cold start BASIC  
J5 cold start RAE

#### RAE LINK:

1. PUT/FILENAME save source file
2. GET/FILENAME load source file  
GET/FILENAME A append to source file
3. DC #x change device number  
DC ! read error channel  
DC ? list directory  
DC /DISKCOMMAND send disk command
4. .CT FILENAME continue on disk

#### BASIC LINK:

1. CONTROL C exit to monitor
2. #SP "FILENAME" save program to disk
3. #LP "FILENAME" load program from disk
4. #DC "#x" (same as RAE LINK)  
#DC "!"  
#DC "?"  
#DC "/DISKCOMMAND"

## SYM-1 1541 DOS SYSTEM

### USING THE SYSTEM

After completing the necessary installation steps you are now ready to proceed with testing your System. The normal power up sequence should involve turning on the peripherals and disk drive first and then the SYM-1. Occasionally the error light on the drive will flash, which is not a problem. You might want to reset the SYM-1 which should reset the disk drive, causing it to start momentarily and stop if all is set up properly. The next step is to load the desired object code from tape. For your convenience the object code is supplied on cassette at two or more different start addresses (\$7000 and \$9000). This is not necessary if you have installed the EPROM option. Following a successful tape load please verify the object code and make sure the check sum agrees with the value on your tape directory. If all is OK, use the SUPERMON command G xxxx to cold start the DOS where xxxx is the start address of the object code. The drive will become active, briefly, and the error light may flash, but not always. To clear the error channel use the command: SC !.

The command format is designed to be simple and similar for each of the software links. In reading this section please refer to the Command Summary section for the correct command format. One of the first steps required to use the System is to "NEW" (format) a blank disk. This is accomplished by sending the 1541 disk NEW command as described in the 1541 manual. The advantage of using the SYM-1 1541 DOS commands is that the procedure is greatly simplified, since the CBM BASIC commands, OPEN, PRINT, and CLOSE are not required. They are an integral part of the single command: SC /NØ:Disk Name,ID#. Other disk commands may be sent in a similar manor, such as: initialize (SC /I), validate (SC /V), or rename a file (SC /RØ:Newname=Oldname) etc.. This format is also used by the RAE and BAS Links with the appropriate command (ie DC /... or #DC "/...", respectively).

#### A. USING THE MONITOR LINK.

When using the Monitor Link, the Load and Save commands require the memory addresses to be entered in hexadecimal numbers similar to other SUPERMON commands. The Save command requires that 2 address parameters be

## SYM-1 1541 DOS SYSTEM

entered, while the Load command may have either zero or one parameters depending on whether a relocated Load is desired. The "/" is used to delineate the start of the file name or disk command. In general, file names may contain any ASCII character including spaces. However, spaces are not allowed in file names for the RAE and BAS Links. The reason for permitting spaces in the Monitor Link is to read and/or rename a 1541 compatible disk file, which quite often contain spaces. If a typing error occurs while inputting the file name, it may be reentered in the Monitor Link by typing a control D. A new prompt "/" is provided, permitting entry of the new file name. When using the SC #x, SC !, and SC ? commands no <Return> is required to execute the command. When listing the disk directory, it is possible to interrupt it by typing the Break key. The listing may be continued by hitting the space bar. This can be very useful for viewing long directory listings. However, when you interrupt the listing, the drive will continue to turn, so it is best not to pause for long periods of time. All of the Monitor disk commands are patched through the URSVEC in system RAM. To use software which also requires the URSVEC, a new URSVEC has been provided (NEW.URSV) located at \$A606 in system RAM. It functions the same as the original URSVEC, returning to the Monitor ERMSG routine.

### B. USING THE RAE LINK.

To enter RAE and initialize the RAE Link the Monitor J 5 command is used. Following initialization, the following file parameters are set: Source files \$200-\$5FFC, Label files \$6000-\$6EFC, and Relocation Buffer \$6F00. The command format for the RAE Link is similar to the Monitor Link with some important differences. The GET and PUT commands are used to load and save RAE source files to disk. These commands utilize the RAE GET and PUT vectors normally used for cassette tape input and output. When the RAE Link is initialized, these vectors are set to the disk routines, with the default device number set to 8. To use the normal cassette GET and PUT commands the device number must be set to zero (DC #0). The form for these commands are: GE/Filename and PU/Filename. There is no need to enter a space after the GE or PU, since the "/" delineates the start of the file name. No spaces are permitted in file names. It is also possible to append a source file from disk to one in memory, the format is: GE/Filename A. When saving a very large source file to disk there may be a slight delay before the drive starts to save the data as certain parameters are set up. The other disk commands are implemented through the RAE DC command, using the DC



## SYM-1 1341 DOS SYSTEM

vector. To aid in future enhancements, a new DC vector (NEW.DCV) has been provided in system RAM at \$A610. All returns to RAE after a DC command are vectored through the new DC vector, which provides a return. To assemble source files from disk the command .CT Filename is used as the pseudo op directive.

It is possible to alter the RAE Link initialization sequence by altering the RAE.SETUPV located at \$A600 in system RAM. Normally when RAE is entered via a Monitor J 5 command, a Monitor Execute command occurs which vectors the program flow through the RAE.SETUPV and then to RAE.SETUP. By changing the RAE.SETUPV to a user supplied routine, RAE may be configured differently. It is important to note that the GET, PUT, and DC vectors set up procedure should not be altered.

### C. USING THE BASIC LINK.

To enter BASIC and initialize the BAS Link, the Monitor J 0 command is used. Following initialization the BASIC HIMEM is set to 28672 (\$7000), the column limit to 80, and the device number is set to 8. The BASIC software link provides the same features as the Monitor and RAE software links. All commands are patched through OUTVEC except the exit to Monitor command (control C) which is linked via INVEC. Similar to the other links, a new INVEC (NEW.INV, \$A60C) and a new OUTVEC (NEW.OUTV, \$A609) are provided in system RAM.

As with the RAE Link, it is possible to alter the BAS Link initialization sequence. Normally when BASIC is entered via a Monitor J 0 command, program flow goes to BAS.INIT which performs a Monitor Execute command. Program flow is then directed through the BAS.COLDV and subsequently to the BAS.COLD routine. By changing the BAS.COLDV located at \$A603 in system RAM to a user supplied routine, the BASIC initialization parameters may be changed.

## SYM-1 1541 DOS SYSTEM

### ERROR CODES

Many of the more common errors which occur are handled by the 1541 disk drive DOS. An error condition is noted by the flashing red light on the drive. To find which error actually occurred, the error channel may be read. Some errors are handled by the SYM-1 1541 DOS routines. The Primitive routines set a bit in the status register depending on the type of error. When the error occurs, the status register is displayed. The significance of the bits are as follows:

---

Status Bit	Hex Value	Description
1000 0000	80	device not present
0100 0000	40	EOI
0010 0000	20	ATN error
0001 0000	10	not used
0000 1000	8	not used
0000 0100	4	not used
0000 0010	2	Read time out
0000 0001	1	Send time out

The Monitor Link handles all of its errors using the typical Monitor method of displaying the accumulator. Within the RAE Link the following additional error codes may be encountered:

- 31 - Text file overflow
- 32 - Device number error
- 33 - Disk I/O error
- with status register displayed
- 34 - File name error

Similar to the RAE Link, the BAS Link handles a few errors internally. The error codes are:

- 31 - Device number error
- 32 - File name error
- 33 - Load error
- 34 - Save error
- 35 - Text overflow error
- 36 - Disk command error

# SYM-1 1541 DOS SYSTEM

## TABLE OF SYSTEM RAM VECTORS

1. RAE.SETUPV	\$A600	Used to set up and start RAE following a monitor J 5 command.
2. BAS.COLDV	\$A603	Used to set up and cold start BASIC following a monitor J 0 command.
3. NEW.URSV	\$A606	The new monitor URS vector which maybe user patched for more monitor commands.
4. NEW.OUTV	\$A609	The BASIC Link patches in via OUTVEC and moves the routine address (TOUT) to NEW.OUTV.
5. NEW.INV	\$A60C	The BASIC Link patches in via INVEC and moves the routine address to NEW.INV
6. NEW.DCV	\$A610	The RAE DC vector is used to patch all disk utility commands. The NEW.DCV allows the addition of even more commands.
7. ACC.VEC	\$A613	A vector for future use. The high address of the running version of DOS is saved so that utilities may vector to the system routines via the jump table.

# SYM-1 1541 DOS SYSTEM

## TABLE OF SYSTEM AND PRIMITIVE VECTORS

### SYSTEM VECTORS

1. LOADV	\$xx03	Load program to RAM. P1= device #, P2=reloc. addr. P3=reloc. flag.
2. SAVEV	\$xx06	Save RAM to disk. P1=device #, P2=start addr., P3=end addr.
3. DISKCMDV	\$xx09	Send 1541 disk command. Command sequence in FNAME, file name length in FN.LEN, and device # in C.DEV.
4. DISK.STV	\$xx0C	Read disk error/status channel.
5. DIR.ENTV	\$xx0F	Display directory. Device # in Accum. when called.
6. GETNAMEV	\$xx12	Monitor Link get file name and length routine.
7. DISKCLOSEV	\$xx15	Open channel for I/O, SET. LISTN, and UNLISTN.
8. DISKOPENV	\$xx18	Open channel for I/O, SET. LISTN, send file name, and UNLISTN.
9. SETUPVIAV	\$xx1B	Initfalize and set up VIA #1.

### PRIMITIVE VECTORS

10. TALKV	\$xx1E	Command serial device to TALK.
11. LISTENV	\$xx21	Command serial device to LISTEN.
12. SECONDV	\$xx24	Send LISTEN/secondary addr.
13. TALKSAV	\$xx27	Send TALK/secondary addr.
14. CIOUTV	\$xx2A	Out byte to serial port.
15. UNTALKV	\$xx2D	Tell serial bus to UNTALK.
16. UNLISTENV	\$xx30	Tell serial bus to UNLISTEN.
17. ACPTRV	\$xx33	Input byte from serial bus.

## SYM-1 1541 DOS SYSTEM

### UTILITIES AND ENHANCEMENTS

The basic SYM-1 1541 DOS provides a flexible foundation for the addition of future commands and utilities. Many commands are possible such as Append to BASIC programs, a RUN command to load and run a program, and OPEN and CLOSE commands to write data to disk. Utilities could be used to copy diskettes and to read/write a disk sector. Many new enhancements and utilities will become available in the near future at a very reasonable cost.

### DISCLAIMER

The SYM-1 1541 DOS System has been extensively tested and is guaranteed to function as described. It was designed to be as flexible as possible and to accommodate system differences and individual software enhancements. However, it is possible that some software may require changes to fully utilize the DOS System. If and when problems are encountered, please contact me and I will attempt to find a solution.

I personally guarantee that each purchaser will receive a functioning hardware interface module and a readable copy of the DOS object code on cassette. If either part is found to be defective, please return that part within 90 days from the date of purchase and it will be replaced free of charge. No other guarantees are either expressed or implied and the user assumes all responsibility for the use and suitability of this System for his or her applications.

### COPYRIGHT NOTICE

COPYRIGHT (C) 1984 by Ronald A. Jordan

This copy of the SYM-1 1541 DOS SYSTEM is provided for the personal use and enjoyment of the purchaser. Reproduction of any portion of this manual or software by any means whatsoever without the express written permission of the author is prohibited.

SYM-1 1541 DOS SYSTEM

All prices include shipping and handling unless otherwise stated. Please allow 4-6 weeks for delivery. Overseas orders add \$10.00.

- (1) SYM-1541 DOS \$95.00
- (2) DOS - Special I/O config. (add \$25.00)
- (3) EPROM option (add \$15.00)
- (4) Source files on disk or cassette \$25.00

Address all mail orders and communications to:

JORDAN & ASSOCIATES  
2611 Madrono Drive  
Ann Arbor, MI 48103

For additional information, telephone  
on weekdays, 6:00 PM-9:00 PM EST, or  
weekends, 9:00 AM-6:00 PM.  
(313) 663-6374

```

0005 ; SYM-1 INTERFACE ROUTINES FOR THE COMMODORE 1541
0010 ;     DISK DRIVE - VERSION 1.0
0015 ;
0020 ;     COPYRIGHT 1984 BY
0025 ;     RONALD A. JORDAN
0030 ;     2611 MADRONO DRIVE
0035 ;     ANN ARBOR, MI 48103
0040 ;
0045 .BA $7000
0050 .OS
0055 .CE
0060 ;
0065 DISK.CHR .DE $F8 ;DISK INPUT CHR BUILD
0070 BIT.CNT .DE $F9 ;DISK BIT COUNT
0075 STATUS .DE $FC ;DISK ROUTINE STATUS
0080 CMD.CHR .DE STATUS
0085 EOI.FLG .DE $FD ;DISK EOI FLAG
0090 CURAD .DE $FE ;CURRENT ADDRESS
0095 ;
0100 FNAME .DE $135 ;FILE NAME
0105 DF.CHR .DE $153 ;DEFERRED CHAR
0110 FN.LEN .DE $154 ;FILE NAME LENGTH
0115 ;
0120 ; MONITOR ROUTINES
0125 USRENT .DE $8035
0130 ERMSG .DE $8171
0135 PSHOVE .DE $8208
0140 ASCNIB .DE $8275
0145 INCP3 .DE $8293
0150 P2SCR .DE $829C
0155 INCCMP .DE $82B2
0160 SPACE .DE $8342
0165 INSTAT .DE $8386
0170 CRLF .DE $834D
0175 EXECUTE .DE $8855+
0180 INCHR .DE $8A1B
0185 OUTCHR .DE $8A47
0190 INTCHR .DE $8A5B
0195 TOUT .DE $8AA0
0200 ACCESS .DE $8B86
0205 ;
0210 ; SYSTEM RAM
0215 RAE.SETUPV .DE $A600 ;RAE SETUP VECTOR
0220 BAS.COLDV .DE $A603 ;BAS COLD START VECTOR
0225 NEW.URSV .DE $A606 ;NEW MON. URS VECTOR
0230 NEW.OUTV .DE $A609 ;NEW CHR OUT VECTOR
0235 NEW.INV .DE $A60C ;NEW CHR IN VECTOR
0240 NEW.DCV .DE $A610 ;NEW RAE DC VECTOR
0245 ACC.VEC .DE $A613 ;COMMON ACCESS VECTOR
0250 J0.VEC .DE $A620 ;BASIC COLD START
0255 J4.VEC .DE $A628 ;RAE WARM START
0260 J5.VEC .DE $A62A ;RAE COLD START
0265 DF.FLG .DE $A62E ;SERIAL DEFERRED FLAG
0270 SA.CMD .DE $A62F ;SECONDARY ADDRESS/COMMAND
0275 SCRA .DE $A63A
0280 DISK.DEV .DE $A647 ;DISK DEVICE NUMBER
0285 APPFLG .DE $A648 ;APPEND FLAG

```



```

0290 PARNR      .DE $A649
0295 P3         .DE $A64A
0300 P2         .DE $A64C
0305 P1         .DE $A64E
0310 C.DEV      .DE P1
0315 TECHD      .DE $A653
0320 TOUTFL      .DE $A654
0325 LASTCMD     .DE $A657
0330 INVEC       .DE $A660
0335 OUTVEC      .DE $A663
0340 URSVEC      .DE $A669
0345 ;
0350 ; SERIAL BUS VIA REFERENCES
0355 VIA1DDRA    .DE $A003      ;VIA #1 DATA DIRECTION - PORT A
0360 VIA1T2L     .DE $A008      ;VIA #1 TIMER 2 LOW ORDER
0365 VIA1T2H     .DE $A009      ;VIA #1 TIMER 2 HIGH ORDER
0370 VIA1PCR     .DE $A00C      ;VIA #1 PERIPHERAL CTL REG
0375 VIA1IFR     .DE $A00D      ;VIA #1 INTERRUPT FLAG REG
0380 VIA1DRA     .DE $A00F      ;VIA #1 PORT A, NO HANDSHAKE
0385 ;
0390 ; JUMP TABLES
7000- 4C 36 70 0395      JMP START
0400 ; SYSTEM VECTORS
7003- 4C 0E 71 0405 LOADV  JMP LOAD
7006- 4C 83 71 0410 SAVEV  JMP SAVE
7009- 4C C9 71 0415 DISKCMDV JMP DISK.CMD
700C- 4C E2 71 0420 DISKSTV JMP DISK.ST
700F- 4C 3D 72 0425 DIRENTV JMP DIR.ENTRY
7012- 4C 2B 73 0430 GETNAMEV JMP GET.NAME
7015- 4C 44 73 0435 DISKCLOSEV JMP DISK.CLOSE
7018- 4C 56 73 0440 DISKOPENV JMP DISK.OPEN
701B- 4C 75 73 0445 SETUPVIAV JMP SETUP.VIA
0450 ;
0455 ; PRIMITIVE VECTORS
701E- 4C A9 73 0460 TALKV   JMP TALK
7021- 4C AC 73 0465 LISTENV  JMP LISTEN
7024- 4C 5B 74 0470 SECONDV  JMP SECOND
7027- 4C 6A 74 0475 TALKSAV  JMP TALK.SA
702A- 4C 81 74 0480 CIOUTV   JMP CIOUT
702D- 4C 96 74 0485 UNTALKV   JMP UNTALK
7030- 4C A5 74 0490 UNLISTENV JMP UNLISTEN
7033- 4C BA 74 0495 ACPTRV   JMP ACPTR
0500 ;
0505 ;
0510 ;
0515 ;
=====
INITIALIZATION ROUTINES
=====
7036- 20 1B 70 0520 START   JSR SETUPVIAV      ; INITIALIZE VIA
7039- A9 08      0525      LDA #8
703B- 8D 47 A6   0530      STA DISK.DEV        ; DRIVE #8 SELECTED
703E- A9 4C      0535 VEC.SETUP LDA #$4C        ; SET SOME JUMPS
7040- 8D 00 A6   0540      STA RAE.SETUPV
7043- 8D 03 A6   0545      STA BAS.COLDV
7046- 8D 06 A6   0550      STA NEW.URSV
7049- 8D 09 A6   0555      STA NEW.OUTV
704C- 8D 0C A6   0560      STA NEW.INV
704F- 8D 10 A6   0565      STA NEW.DCV
7052- 8D 13 A6   0570      STA ACC.VEC          ; SETUP COMMON ACCESS VEC
7055- A9 70      0575      LDA #H,START        ; HIGH ADDR OBJ CODE START

```



```

7057- 8D 15 A6 0580 STA ACC.VEC+2 ;SAVE IN COMMON ACC. VEC
705A- A2 D1 0585 LDX #D1 ;SETUP NEW URS RETURN VEC
705C- A9 81 0590 LDA #S1
705E- 8E 07 A6 0595 STX NEW.URSV+1
7061- 8D 08 A6 0600 STA NEW.URSV+2
7064- A2 AE 0605 LDX #L,NEW.CMDS ;PATCH MON. EXTENSIONS
7066- A9 70 0610 LDA #H,NEW.CMDS
7068- 8E 6A A6 0615 STX URSVEC+1
706B- 8D 68 A6 0620 STA URSVEC+2
706E- A2 00 0625 LDX #L,RAE.INIT ;SET J5 FOR RAE
7070- A9 78 0630 LDA #H,RAE.INIT
7072- 8E 2A A6 0635 STX J5.VEC
7075- 8D 2B A6 0640 STA J5.VEC+1
7078- A2 10 0645 LDX #L,RAE.SETUP
707A- A9 78 0650 LDA #H,RAE.SETUP
707C- 8E 01 A6 0655 STX RAE.SETUPV+1
707F- 8D 02 A6 0660 STA RAE.SETUPV+2
7082- A2 03 0665 LDX #L,RAE.WARM ;SET J4 RAE WARM START
7084- A9 B0 0670 LDA #H,RAE.WARM
7086- 8E 28 A6 0675 STX J4.VEC
7089- 8D 29 A6 0680 STA J4.VEC+1
708C- A2 27 0685 LDX #L,BAS.INIT ;SET J0 FOR BASIC
708E- A9 7A 0690 LDA #H,BAS.INIT
7090- 8E 20 A6 0695 STX J0.VEC
7093- 8D 21 A6 0700 STA J0.VEC+1
7096- A2 37 0705 LDX #L,BAS.COLD
7098- A9 7A 0710 LDA #H,BAS.COLD
709A- 8E 04 A6 0715 STX BAS.COLDV+1
709D- 8D 05 A6 0720 STA BAS.COLDV+2
70A0- A2 00 0725 LDX #0
70A2- 8D 70 75 0730 INIT.MSG LDA MESSAGE,X ;OUTPUT MESSAGE
70A5- F0 06 0735 BEQ END.MSG
70A7- 20 47 8A 0740 JSR OUTCHR
70AA- E8 0745 INX
70AB- D0 F5 0750 BNE INIT.MSG ;ALWAYS, MORE MESSAGE
70AD- 60 0755 END.MSG RTS
0760 ;
0765 ;
0770 ;
0775 ;
=====
NEW COMMANDS - VECTORED VIA THE MONITOR URSVEC
=====
70AE- 85 FC 0780 NEW.CMDS STA *CMD.CHR ;SAVE LAST CHAR
70B0- AD 57 A6 0785 LDA LASTCMD ;GET LAST COMMAND
70B3- C9 13 0790 L2/CHK CMP #S13 ;DISK LOAD?
70B5- D0 09 0795 BNE S2/CHK ;...NO, TRY SAVE
70B7- A5 FC 0800 LDA *CMD.CHR
70B9- C9 2F 0805 CMP #' / ;NAME DELIMINATOR?
70BB- D0 32 0810 BNE EXT.CMDS ;...NO, ERROR
70BD- 4C F2 70 0815 JMP L2
0820 ;
70C0- C9 1E 0825 S2/CHK CMP #S1E ;DISK SAVE?
70C2- D0 09 0830 BNE SC.CHK ;...NO, TRY DISK CMDS
70C4- A5 FC 0835 LDA *CMD.CHR
70C6- C9 2F 0840 CMP #' /
70C8- D0 25 0845 BNE EXT.CMDS ;...NO, ERROR
70CA- 4C 70 71 0850 JMP S2
0855 ;
70CD- C9 1F 0860 SC.CHK CMP #S1F ;SC?
70CF- D0 1E 0865 BNE EXT.CMDS ;...NO, ERROR

```

```

70D1- A5 FC      0870      LDA *CMD.CHR          ;GET LAST CHAR
70D3- C9 2F      0875      CMP #'/'          ;DISK COMMAND?
70D5- D0 03      0880      BNE ST.CHK          ;...NO, TRY STATUS
70D7- 4C BC 71    0885      JMP SC
                        0890 ;
70DA- C9 21      0895 ST.CHK      CMP #'!'          ;STATUS?
70DC- D0 03      0900      BNE D.R.CHK        ;...NO, TRY DIRECTORY
70DE- 4C D8 71    0905      JMP ST
                        0910 ;
70E1- C9 3F      0915 DIR.CHK     CMP #'?'          ;DIRECTORY?
70E3- D0 03      0920      BNE NU.CHK        ;...NO, TRY DEVICE #
70E5- 4C 36 72    0925      JMP DIR
                        0930 ;
70E8- C9 23      0935 NU.CHK     CMP #'#'          ;DEVICE #?
70EA- D0 03      0940      BNE EXT.CMDS      ;NO MORE COMMANDS
70EC- 4C 12 73    0945      JMP NU
                        0950 ;
70EF- 4C 06 A6    0955 EXT.CMDS   JMP NEW.URSV      ;NO MORE COMMANDS
                        0960 ;
                        0965 ;
                        0970 ;
                        0975 ;
                        =====
                        LOAD FROM DISK (L2 - MONITOR COMMAND)
                        =====
70F2- 20 08 82    0980 L2        JSR PSHOVE
70F5- AE 49 A6    0985      LDX PARNR          ;GET # PARMS
70F8- F0 0B      0990      BEQ L2/0PARM
70FA- E0 01      0995      CPX #1             ;CHK FOR 1
70FC- F0 04      1000      BEQ L2/1PARM       ;...1 PARM OK
                        1005 ;
70FE- A9 13      1010 L.ERROR     LDA #*13
7100- 38          1015      SEC               ;SET ERROR FLAG
7101- 60          1020      RTS
                        1025 ;
7102- 8E 4A A6    1030 L2/1PARM   STX P3         ;SET RELOCATE FLAG NON-0
7105- AD 47 A6    1035 L2/0PARM   LDA DISK.DEV
7108- 8D 4E A6    1040      STA C.DEV         ;DEVICE # TO P1
710B- 20 12 70    1045      JSR GETNAMEV      ;GET FILE NAME
                        1050 ;
                        1055 ;SERIAL LOAD ROUTINES
                        1060 ;=====
                        1065 ;      P1-DEVICE NUMBER      P2-RELOCATE ADDRESS (OPT.)
                        1070 ;      P3-RELOCATE FLAG (NON ZERO = RELOCATE)
                        1075 ;      FNAME AND FN.LEN SAME AS IN SAVE ROUTINES
                        1080 ;
710E- 20 18 70    1085 LOAD      JSR SETUPVIAV    ;UNLOCK RAM & SET VIA
7111- A9 F0      1090      LDA #*F0          ;OPEN/LOAD SA
7113- 20 18 70    1095      JSR DISKOPENV     ;DO DISK OPEN
7116- B0 4F      1100      BCS ERR.RTN        ;...ERROR
7118- A9 60      1105      LDA #*60          ;SETUP LOAD SA
711A- 20 8D 73    1110      JSR SET.TALK      ;TELL TO TALK
711D- 20 28 71    1115      JSR RCV.PGM       ;GET PROGRAM DATA
7120- 20 2D 70    1120      JSR UNTALKV
7123- A9 E0      1125      LDA #*E0          ;CLOSE/LOAD SA
7125- 4C 15 70    1130      JMP DISKCLOSEV
                        1135 ;
                        1140 ;RECEIVE PROGRAM DATA FROM SERIAL BUS
                        1145 ;=====
7128- 20 33 70    1150 RCV.PGM   JSR ACPTRV      ;START ADR LOW
712B- 85 FE      1155      STA *CURAD

```

```

712D- A5 FC      1160      LDA #STATUS          ;CHECK FOR TIMEOUT
712F- 29 02      1165      AND #02
7131- D0 34      1170      BNE ERR.RTN          ;...TIMEOUT ERROR
7133- 20 33 70   1175      JSR ACPTRV          ;START ADR HIGH
7136- 85 FF      1180      STA #CURAD+1
7138- AD 4A A6   1185      LDA P3              ;CHECK RELOCATE FLAG
713B- F0 03      1190      BEQ =+4            ;...NO RELOCATE
713D- 20 9C 82   1195      JSR P2SCR          ;ELSE USE NEW START ADR
                        1200 ;
7140- A9 FD      1205 LOAD.BYTE LDA #FD          ;TURN OFF TIMEOUT BIT
7142- 25 FC      1210      AND #STATUS
7144- 85 FC      1215      STA #STATUS
7146- AD 0F A0   1220      LDA VIA1DRA          ;IS ATN STILL CLEAR?
7149- 29 04      1225      AND #04
714B- F0 1E      1230      BEQ ATN.ERR          ;...SOMEBODY ELSE HAS ATN
714D- 20 33 70   1235      JSR ACPTRV          ;GET A BYTE
7150- AA         1240      TAX                  ;SAVE
7151- A5 FC      1245      LDA #STATUS
7153- 29 02      1250      AND #02
7155- D0 E9      1255      BNE LOAD.BYTE        ;CHK FOR TIMEOUT
7157- BA         1260      TXA                  ;...IGNORE IF SO
7158- 91 FE      1265      STA (CURAD),Y
715A- E6 FE      1270      INC #CURAD          ;NEXT PLACE TO STORE
715C- D0 02      1275      BNE =+3
715E- E6 FF      1280      INC #CURAD+1
7160- 24 FC      1285      BIT #STATUS          ;TEST FOR EOI
7162- 50 DC      1290      BVC LOAD.BYTE        ;...NOT YET
7164- 84 FC      1295      STY #STATUS          ;CLEAR STATUS
7166- 60         1300      RTS
                        1305 ;
                        1310 ;COMMON ERROR RETURN
                        1315 ;=====
7167- A5 FC      1320 ERR.RTN LDA #STATUS          ;USE STATUS AS ERR
7169- 38         1325 S.ERROR SEC                ;SET FLAG
716A- 60         1330      RTS
                        1335 ;
                        1340 ;ATTENTION ERROR
                        1345 ;=====
716B- A9 20      1350 ATN.ERR LDA #020          ;INVALID ATN STATUS
716D- 4C 6A 75   1355      JMP SET.ST
                        1360 ;
                        1365 ;
                        1370 ;
                        1375 ;
                        1380 S2
                        1385      LDX PARNR          ;NUMBER OF PARAMETERS
7173- E0 02      1385      CPX #2              ;MUST BE 2 PARMS
7175- D0 F2      1390      BNE S.ERROR          ;...NOT 2
7177- 20 12 70   1395      JSR GETNAMEV        ;GET NAME
717A- 20 93 82   1400      JSR INCP3           ;BUMP P3 TO END+1
717D- AD 47 A6   1405      LDA DISK.DEV
7180- 8D 4E A6   1410      STA C.DEV
                        1415 ;
                        1420 ;SERIAL SAVE ROUTINES
                        1425 ;=====
                        1430 ;
                        1435 ;
                        1440 ;
                        1445 ;
                        P1-DEVICE NUMBER          P2-START ADDRESS
                        P3-END ADDRESS+1         NAME @ FNAME (MAX 30 CHARS)
                        NAME LENGTH @ FN.LEN

```

```

7183- 20 1B 70 1450 SAVE      JSR SETUPVIAV
7186- 20 9C 82 1455      JSR P2SCR      ;START ADR TO PAGE ZERO
7189- A9 F1 1460      LDA #F1          ;OPEN/SAVE SA
718B- 20 18 70 1465      JSR DISKOPENV  ;DISK OPEN
718E- B0 D7 1470      BCS ERR.RTN      ;...ERROR
7190- A9 61 1475      LDA #61          ;SETUP SAVE SA
7192- 20 9B 73 1480      JSR SET.LSTN
7195- 20 A0 71 1485      JSR SEND.PGM    ;USE COMMON SEND
7198- 20 30 70 1490      JSR UNLISTENV   ;TELL DEVICE WE ARE DONE
719B- A9 E1 1495      LDA #E1          ;CLOSE/SAVE SA
719D- 4C 15 70 1500      JMP DISKCLOSEV
1505 ;
1510 ;SEND PROGRAM TO SERIAL BUS
1515 ;=====
71A0- A5 FE 1520 SEND.PGM LDA *CURAD      ;START WITH ADR
71A2- 20 2A 70 1525      JSR CIOUTV
71A5- A5 FF 1530      LDA *CURAD+1
71A7- 20 2A 70 1535      JSR CIOUTV
1540 ;
71AA- AD 0F A0 1545 SAVE.BYTE LDA VIA1DRA
71AD- 29 04 1550      AND #04          ;CHECK ATN IN
71AF- F0 BA 1555      BEQ ATN.ERR      ;...SOMEBODY ELSE HAS IT
71B1- B1 FE 1560      LDA (CURAD),Y
71B3- 20 2A 70 1565      JSR CIOUTV
71B6- 20 B2 82 1570      JSR INCCMP     ;BUMP TO NEXT BYTE
71B9- D0 EF 1575      BNE SAVE.BYTE    ;...LOOP, MORE TO DO
71BB- 60 1580      RTS
1585 ;
1590 ;
1595 ;
1600 ;
1605 SC
71BC- E0 00 1610      CPX #0          ;ONLY 0 PARM
71BE- D0 A9 1615      BNE S.ERROR      ;...ONLY 0 ALLOWED
71C0- AD 47 A6 1615      LDA DISK.DEV
71C3- 8D 4E A6 1620      STA C.DEV
71C6- 20 12 70 1625      JSR GETNAMEV
71C9- A9 FF 1630 DISK.CMD LDA #FF      ;OPEN COMMAND CHANNEL
71CB- 20 18 70 1635      JSR DISKOPENV
71CE- B0 99 1640      BCS S.ERROR
71D0- A9 EF 1645      LDA #EF          ;CLOSE COMMAND CHANNEL
71D2- 20 15 70 1650      JSR DISKCLOSEV
71D5- B0 92 1655      BCS S.ERROR
71D7- 60 1660      RTS                ;GOOD RETURN
1665 ;
1670 ;
1675 ;
1680 ;
1685 ;(DOES NOT OPEN/CLOSE CMD CHANNEL AS C-64 WEDGE)
1690 ;
1695 ST
71D8- E0 00 1700      CPX #0
71DA- D0 8D 1705      BNE S.ERR        ;...ONLY 0 ALLOWED
71DC- AD 47 A6 1710      LDA DISK.DEV
71DF- 8D 4E A6 1715      STA C.DEV
71E2- A9 35 1720 DISK.ST LDA #L,FNAME  ;SET ADR TO RECEIVE
71E4- 85 FE 1725      STA *CURAD
71E6- A9 01 1730      LDA #H,FNAME
71E8- 85 FF 1735      STA *CURAD+1
71EA- A9 1E 1735      LDA #30          ;30 CHRS MAX

```

```

71EC- 8D 54 01 1740 STA FN.LEN
71EF- A9 6F 1745 LDA #6F ;USE COMMAND CHANNEL
71F1- 20 08 72 1750 JSR RCV.LINE
71F4- B0 11 1755 BCS ST.RTS ;...RETURN, CARRY SET
71F6- A0 00 1760 LDY #0 ;OUTPUT STATUS RECEIVED
71F8- 20 4D 83 1765 JSR CRLF ;START ON NEXT LINE
71FB- B1 FE 1770 ST.LP LDA (CURAD),Y
71FD- 20 47 8A 1775 JSR OUTCHR
7200- C8 1780 INY
7201- CC 54 01 1785 CPY FN.LEN ;TO END?
7204- D0 F5 1790 BNE ST.LP ;...MORE CHRS TO OUTPUT
7206- 18 1795 CLC
7207- 60 1800 ST.RTS RTS
1805 ;
1810 ;RECEIVE LINE INTO (CURAD)
1815 ;=====
1820 ; UNTIL 1) A C/R
1825 ; 2) EOI
1830 ; 3) FN.LEN CHRS RECEIVED
1835 ; FL.LNG RETURNED WITH # CHRS RECEIVED
1840 ;
7208- 20 1B 70 1845 RCV.LINE JSR SETUPVIAV ;SETUP VIA
720B- A0 00 1850 LDY #0 ;CLEAR STATUS
720D- 84 FC 1855 STY *STATUS
720F- 20 8D 73 1860 JSR SET.TALK
7212- D0 17 1865 BNE RCV.ERR ;...BAD STATUS
7214- 20 33 70 1870 RC.CHR JSR ACPTRV ;GET A CHR
7217- 91 FE 1875 STA (CURAD),Y ;STORE IT
7219- C9 0D 1880 CMP #0D ;C/R?
721B- F0 11 1885 BEQ RCV.END ;...NORMAL END
721D- C8 1890 INY ;COUNT CHR
721E- CC 54 01 1895 CPY FN.LEN ;TO MAX?
7221- F0 0B 1900 BEQ RCV.END ;...NORMAL END
7223- A5 FC 1905 LDA *STATUS ;CHK STATUS
7225- F0 ED 1910 BEQ RC.CHR ;...NORMAL, LOOP
7227- C9 40 1915 CMP #40 ;EOI ONLY?
7229- F0 03 1920 BEQ RCV.END ;...NORMAL END
722B- 4C 67 71 1925 RCV.ERR JMP ERR.RTN ;LOAD ST & RETURN W/CARRY
722E- 8C 54 01 1930 RCV.END STY FN.LEN ;STORE RETURNED LENGTH
7231- 20 2D 70 1935 JSR UNTALKV ;...DONE
7234- 18 1940 CLC
7235- 60 1945 RTS
1950 ;
1955 ;
1960 ; LOAD/DISPLAY DIRECTORY (SC ? - MON COMMAND)
1965 ;
7236- E0 00 1970 DIR CPX #0
7238- D0 A0 1975 DIR.ERR BNE ST.ERR ;...ONLY 0 ALLOWED
723A- AD 47 A6 1980 LDA DISK.DEV
723D- 8D 4E A6 1985 DIR.ENTRY STA C.DEV ;RAE & BAS DIRECT. START
7240- A9 24 1990 LDA #'$ ;DIRECTORY NAME
7242- 8D 35 01 1995 STA FNAME
7245- A2 01 2000 LDX #1
7247- 8E 54 01 2005 STX FN.LEN ;NAME LENGTH=1 CHR
724A- A9 F0 2010 LDA #F0 ;'LOAD' SA
724C- 20 18 70 2015 JSR DISKOPENV
724F- B0 B6 2020 BCS ST.RTS ;...RETURN WITH CARRY SET
7251- A9 60 2025 LDA #60

```



```

7253- 20 8D 73 2030 JSR SET.TALK ;TELL DISK TO TALK
2035 ;
7256- 20 4D 83 2040 JSR CRLF
7259- A9 00 2045 LDA #0
725B- 85 FC 2050 STA *STATUS
725D- A0 03 2055 LDY #3
725F- 8C 54 01 2060 DIR.LINE STY FN.LEN
7262- 20 33 70 2065 JSR ACPTRV
7265- 8D 35 01 2070 STA FNAME
7268- 20 33 70 2075 JSR ACPTRV
726B- 8D 36 01 2080 STA FNAME+1
726E- A5 FC 2085 LDA *STATUS
7270- D0 49 2090 BNE EOI.CHK
7272- AC 54 01 2095 LDY FN.LEN
7275- B8 2100 DEY
7276- D0 E7 2105 BNE DIR.LINE
7278- 20 CE 72 2110 JSR HEX2AD ;CONVERT TO ASCII DEC.
727B- 20 42 83 2115 JSR SPACE
727E- 20 42 83 2120 JSR SPACE
7281- 20 33 70 2125 DIR.FIX JSR ACPTRV ;FIX A DISK QUIRK
7284- C9 20 2130 CMP #$20
7286- F0 F9 2135 BEQ DIR.FIX
7288- D0 0B 2140 BNE DCHR.OUT
728A- 20 33 70 2145 DIR.NAME JSR ACPTRV ;GET FILE NAME NOW
728D- A6 FC 2150 LDX *STATUS
728F- D0 2A 2155 BNE EOI.CHK
7291- C9 00 2160 CMP #0
7293- F0 1F 2165 BEQ CRLF.OUT
7295- 20 47 8A 2170 DCHR.OUT JSR OUTCHR
7298- 20 86 83 2175 JSR INSTAT ;CHECK FOR BREAK
729B- 90 ED 2180 BCC DIR.NAME
729D- AD 53 A6 2185 LDA TECHD
72A0- 48 2190 PHA
72A1- A9 40 2195 LDA #$40 ;ECHO AND OUTPUT OFF
72A3- 8D 53 A6 2200 STA TECHD
72A6- 20 1B 8A 2205 WAIT.SPC JSR INCHR ;SPACE TO CONTINUE
72A9- C9 20 2210 CMP #$20
72AB- D0 F9 2215 BNE WAIT.SPC
72AD- 68 2220 PLA ;GET TECHD BACK
72AE- 8D 53 A6 2225 STA TECHD
72B1- 18 2230 CLC
72B2- 90 D6 2235 BCC DIR.NAME
2240 ;
72B4- 20 4D 83 2245 CRLF.OUT JSR CRLF
72B7- A0 02 2250 LDY #2
72B9- D0 A4 2255 BNE DIR.LINE ;ALWAYS
2260 ;
72BB- C9 40 2265 EOI.CHK CMP #$40 ;EOI ONLY?
72BD- D0 50 2270 BNE DIR.ERR2 ;...OTHER DISK ERROR
72BF- 20 4D 83 2275 JSR CRLF
72C2- A9 00 2280 LDA #0
72C4- 85 FC 2285 STA *STATUS ;CLEAR STATUS
72C6- 20 2D 70 2290 JSR UNTALKV ;TELL DISK TO UNTALK
72C9- A9 E0 2295 LDA #$E0 ;CLOSE/LOAD SA
72CB- 4C 15 70 2300 JMP DISKCLOSEV
2305 ;
72CE- A0 00 2310 HEX2AD LDY #0
72D0- A2 00 2315 NXTDIG LDX #0

```

```

72D2- AD 35 01 2320 SUBEM      LDA FNAME
72D5- 38          2325      SEC
72D6- F9 0B 73 2330      SBC SUBTBL,Y
72D9- 8D 35 01 2335      STA FNAME
72DC- AD 36 01 2340      LDA FNAME+1
72DF- C8          2345      INY
72E0- F9 0B 73 2350      SBC SUBTBL,Y
72E3- 90 08      2355      BCC ADDBAK
72E5- 8D 36 01 2360      STA FNAME+1
72E8- EB          2365      INX
72E9- 88          2370      DEY
72EA- 4C D2 72 2375      JMP SUBEM
72ED- 88          2380 ADDBAK  DEY
72EE- AD 35 01 2385      LDA FNAME
72F1- 79 0B 73 2390      ADC SUBTBL,Y
72F4- 8D 35 01 2395      STA FNAME
72F7- BA          2400      TXA
72F8- 09 30      2405      ORA ##30
72FA- 20 47 BA 2410      JSR OUTCHR
72FD- C8          2415      INY
72FE- C8          2420      INY
72FF- C0 04      2425      CPY #4
7301- 90 CD      2430      BCC NXTDIG
7303- AD 35 01 2435      LDA FNAME
7306- 09 30      2440      ORA ##30
7308- 4C 47 BA 2445      JMP OUTCHR
          2450 ;
730B- 64 00      2455 SUBTBL  .SI $0064          ;100
730D- 0A 00      2460      .SI $000A          ;10
          2465 ;
730F- 4C 69 71 2470 DIR.ERR2  JMP S.ERROR
          2475 ;
          2480 ;
          2485 ;
          2490 ;
          =====
          SET DEVICE NUMBER (SC # - MON COMMAND)
          =====
7312- 20 1B BA 2495 NU      JSR INCHR*          ;GET DEVICE NUMBER
7315- 20 75 B2 2500      JSR ASCNIB          ;CONVERT TO HEX
7318- B0 F5      2505      BCS DIR.ERR2          ;...NOT HEX CHR
731A- C9 08      2510      CMP #B
731C- 90 F1      2515      BCC DIR.ERR2          ;< B, ...ERROR
731E- 8D 47 A6 2520      STA DISK.DEV
7321- 18          2525      CLC
7322- 60          2530      RTS
          2535 ;
          2540 ;
          2545 ;
          2550 ;
          =====
          INPUT FILE NAME
          =====
7323- 20 4D B3 2555 CTROLD  JSR CRLF          ;RESTART FILE NAME
7326- A9 2F      2560      LDA #' /          ;SEND PROMPT
7328- 20 47 BA 2565      JSR OUTCHR
          2570 ;
732B- A2 00      2575 GET.NAME  LDX #0
732D- 20 1B BA 2580 GN.CHR  JSR INCHR          ;GET A CHR
7330- 9D 35 01 2585      STA FNAME,X          ;SAVE CHR
7333- F0 0B      2590      BEQ GN.RTS          ;...IT WAS A C/R
7335- C9 04      2595      CMP ##04          ;CONTOL D?
7337- F0 EA      2600      BEQ CTROLD          ;...YES, RESTART FILE NAME
7339- EB          2605      INX          ;COUNT IT

```

```

733A- E0 1E      2610      CPX #30      ;30 IS MAX
733C- F0 ED      2615      BEQ GET.NAME  ;...START OVER IF TOO MANY
733E- 90 ED      2620      BCC GN.CHR    ;...ELSE LOOP FOR MORE
7340- 8E 54 01   2625 GN.RTS  STX FN.LEN  ;SAVE NUMBER CHRS
7343- 60         2630      RTS
                2635 ;
                2640 ;
                2645 ;      COMMON DISK CLOSE ROUTINES
                2650 ;
                2655 ;      ST CHECKED BEFORE THE CLOSE
                2660 ;      SA IS IN 'A' UPON CALL
                2665 ;      ST IS RETURNED IN 'A'
                2670 ;
7344- A4 FC      2675 DISK.CLOSE LDY *STATUS      ;CHECK PREVIOUS STATUS
7346- D0 0B      2680      BNE ERR.RTN2      ;...ERROR
7348- 20 9B 73   2685      JSR SET.LSTN
734B- 20 30 70   2690      JSR UNLISTENV
734E- A5 FC      2695      LDA *STATUS
7350- D0 01      2700      BNE ERR.RTN2      ;...ERROR, SHOULD BE ZERO
7352- 60         2705      RTS              ;RETURN
                2710 ;
7353- 4C 67 71   2715 ERR.RTN2  JMP ERR.RTN
                2720 ;
                2725 ;
                2730 ;      COMMON DISK OPEN ROUTINES
                2735 ;
                2740 ;      ST CLEARED AT ENTRY
                2745 ;      SA IN 'A' AT ENTRY
                2750 ;
7356- A0 00      2755 DISK.OPEN  LDY #0          ;CLEAR STATUS
7358- 84 FC      2760      STY *STATUS
735A- 20 9B 73   2765      JSR SET.LSTN      ;TELL TO LISTEN
735D- D0 F4      2770      BNE ERR.RTN2      ;...BAD, STATUS RETURNED
735F- AD 54 01   2775      LDA FN.LEN        ;ANY NAME/COMMAND
7362- F0 0C      2780      BEQ NO.DATA        ;...NONE
7364- B9 35 01   2785 SEND.NAME  LDA FNAME,Y    ;GET NAME CHR
7367- 20 2A 70   2790      JSR CIOUTV        ;SEND TO DISK
736A- C8         2795      INY
736B- CC 54 01   2800      CPY FN.LEN        ;END OF NAME YET?
736E- D0 F4      2805      BNE SEND.NAME      ;...LOOP
7370- A0 00      2810 NO.DATA  LDY #0          ;CLEAR STATUS
7372- 4C 30 70   2815      JMP UNLISTENV     ;TELL DEVICE DONE
                2820 ;
                2825 ;
                2830 ;      SETUP VIA ROUTINES - SYM VIA #1
                2835 ;
7375- 48         2840 SETUP.VIA  PHA
7376- 20 86 8B   2845      JSR ACCESS
7379- A9 38      2850      LDA #38          ;ATN,CLK,DATA SET AS OUTPUT
737B- 8D 03 A0   2855      STA VIA1DDRA
737E- A9 20      2860      LDA #20          ;SET CLK & ATN HI, DATA LOW
7380- 8D 0F A0   2865      STA VIA1DRA
7383- A9 00      2870      LDA #00
7385- 8D 08 A0   2875      STA VIA1T2L      ;ZERO T2 LATCHES
7388- 8D 09 A0   2880      STA VIA1T2H
738B- 68         2885      PLA
738C- 60         2890      RTS
                2895 ;

```



```

2900 ;
2905 ;
2910 ;
2915 ; (SA IN 'A')
738D- 48 2920 SET.TALK PHA ;SAVE SA
738E- AD 4E A6 2925 LDA C.DEV ;GET DEVICE #
7391- 20 1E 70 2930 JSR TALKV ;COMMAND TO TALK
7394- 68 2935 PLA ;GET SA BACK
7395- 20 27 70 2940 JSR TALKSAV ;SEND TO DEVICE
7398- A4 FC 2945 LDY *STATUS ;LOAD/TEST STATUS
739A- 60 2950 RTS
2955 ;
2960 ;
2965 ;
2970 ;
2975 ; ( SA IN 'A')
739B- 48 2980 SET.LSTN PHA ;SAVE SA
739C- AD 4E A6 2985 LDA C.DEV ;GET DEVICE #
739F- 20 21 70 2990 JSR LISTENV ;COMMAND TO LISTEN
73A2- 68 2995 PLA ;GET SA BACK
73A3- 20 24 70 3000 JSR SECONDV
73A6- A4 FC 3005 LDY *STATUS ;LOAD/TEST STATUS
73A8- 60 3010 RTS
3015 ;
3020 .CT PRIM1541 0

```

```

0001 .EJ
3100 ; SYM-1 INTERFACE ROUTINES FOR THE COMMODORE 1541
3105 ; DISK DRIVE - VERSION 1.0
3110 ;
3115 ;-----
3120 ; DISK PRIMITIVES
3125 ; THE FOLLOWING ROUTINES ARE FROM THE C-64 OR VIC 20 KERNAL
3130 ;-----
3135 ; (ERRORS ARE RETURNED IN THE STATUS REGISTER)
3140 ;
3145 ;
3150 ; SEND 'TALK' TO DEVICE
3155 ;
3160 ; (COMMAND IN 'A', SAME AS C-64 $ED09/VIC 20 $EE14)
73A9- 09 40 3165 TALK ORA #$40 ;SET TALK BIT
73AB- 2C 3170 .BY $2C
3175 ;
3180 ;
3185 ; SEND 'LISTEN' TO DEVICE
3190 ;
3195 ; (COMMAND IN 'A', SAME AS C-64 $ED0C/VIC 20 $EE17)
73AC- 09 20 3200 LISTEN ORA #$20
73AE- 20 60 75 3205 JSR S.CHK.IDL ;VIC 20 @ $F160
3210 ;
3215 ;
3220 ; SEND 'CONTROL'
3225 ;
3230 ; (SAME AS C-64 $ED11/VIC 20 $EE1C)
73B1- 48 3235 CONTROL PHA ;SAVE CHR
73B2- 2C 2E A6 3240 BIT DF.FLG ;TEST DEFERRED CHR FLAG
73B5- 10 0B 3245 BPL GET.DEF ;...NO DEF CHR PENDING
73B7- 38 3250 SEC
73B8- 66 FD 3255 ROR $EOI.FLG ;SET EOI BIT ON
73BA- 20 E3 73 3260 JSR SEND.OUT ;SEND SERIAL CHR
73BD- 4E 2E A6 3265 LSR DF.FLG
73C0- 46 FD 3270 LSR $EOI.FLG ;CLEAR EOI BIT
73C2- 68 3275 GET.DEF PLA ;GET CHR BACK
73C3- 8D 53 01 3280 STA DF.CHR ;SAVE AS DEFERRED CHR
73C6- 78 3285 SEI
73C7- 20 37 75 3290 JSR SRL.1 ;LET DEVICE HAVE DATA LINE
73CA- C9 3F 3295 CMP #$3F ;???, ERROR IN KERNAL
73CC- D0 03 3300 BNE ATN.LOW
73CE- 20 25 75 3305 JSR CLK.ON ;SET CLOCK HIGH
3310 ;
73D1- AD 0F A0 3315 ATN.LOW LDA VIA1DRA
73D4- 09 08 3320 ORA #$08 ;SET 'ATN OUT' (LOW)
73D6- 8D 0F A0 3325 STA VIA1DRA
3330 ;
73D9- 78 3335 BUS.DIR SEI
73DA- 20 2E 75 3340 JSR CLK.OFF ;CHK BUS DIRECTION
73DD- 20 37 75 3345 JSR SRL.1
73E0- 20 53 75 3350 JSR DLY.1MS ;WAIT 1 MSEC
3355 ;
3360 ;
3365 ; SEND SERIAL OUTPUT CHARACTER
3370 ;
3375 ; (SAME AS C-64 $ED40/VIC 20 $EE49)
73E3- 78 3380 SEND.OUT SEI ;DISABLE INTERRUPTS

```

```

73E4- 20 37 75 3385      JSR SRL.1      ;OUTPUT '1'
73E7- 20 49 75 3390      JSR SRL.IC     ;GET INPUT BIT & CLK IN
73EA- 4A              3395      LSR A
73EB- B0 62          3400      BCS DEV.NP    ;...DEVICE NOT PRESENT
73ED- 20 25 75 3405      JSR CLK.ON     ;SET CLOCK OUT HIGH
73F0- 24 FD          3410      BIT $EOI.FLG  ;TEST EOI FLAG
73F2- 10 0C          3415      BPL S.OUT3    ;...NO EOI
                          3420 ; (SIGNAL EOI)
73F4- 20 49 75 3425 S.OUT1 JSR SRL.IC     ;WAIT ALL LISTENERS READY
73F7- 4A              3430      LSR A
73FB- 90 FA          3435      BCC S.OUT1    ;...NOT YET
73FA- 20 49 75 3440 S.OUT2 JSR SRL.IC     ;WAIT LISTENER TO ACK EOI
73FD- 4A              3445      LSR A
73FE- B0 FA          3450      BCS S.OUT2
7400- 20 49 75 3455 S.OUT3 JSR SRL.IC     ;WAIT ALL LISTENERS READY
7403- 4A              3460      LSR A
7404- 90 FA          3465      BCC S.OUT3
7406- 20 2E 75 3470      JSR CLK.OFF    ;START CLOCK PULSE
                          3475 ; (SEND OUT BYTE)
7409- A9 08          3480      LDA #8      ;8 BITS TO OUTPUT
740B- B5 F9          3485      STA $BIT.CNT
740D- AD 0F A0 3490 S.OUT4 LDA VIA1DRA
7410- CD 0F A0 3495      CMP VIA1DRA    ;MAKE SURE NOT CHANGING
7413- D0 FB          3500      BNE S.OUT4    ;...CHANGING, TRY AGAIN
7415- 4A              3505      LSR A
7416- 4A              3510      LSR A
7417- 90 39          3515      BCC TIME.OUT ;...SET TIMEOUT STATUS
7419- 6E 53 01 3520      ROR DF.CHR     ;GET BIT TO SEND
741C- B0 05          3525      BCS BIT.ON   ;...BIT WAS ON
741E- 20 40 75 3530      JSR SRL.0      ;OUTPUT '0'
7421- D0 03          3535      BNE S.WAIT   ;...UNCONDITIONAL
7423- 20 37 75 3540 BIT.ON JSR SRL.1      ;OUTPUT '1'
7426- 20 25 75 3545 S.WAIT JSR CLK.ON     ;END CLOCK PULSE
7429- EA EA EA 3550      .BY $EA $EA $EA $EA ;C64 NEEDS 60 USEC
742C- EA
742D- AD 0F A0 3555      LDA VIA1DRA
7430- 29 DF          3560      AND #$DF    ;DATA OUT HIGH
7432- 09 10          3565      ORA #$10    ;START NEXT CLOCK CYCLE
7434- BD 0F A0 3570      STA VIA1DRA
7437- C6 F9          3575      DEC $BIT.CNT ;COUNT THE BIT SENT
7439- D0 D2          3580      BNE S.OUT4    ;LAST BIT?, ...GET MORE
                          3585 ; (FRAME HANDSHAKE)
743B- A9 04          3590      LDA #$04    ;*** TIMING ***
743D- 8D 09 A0 3595      STA VIA1T2H    ;START T2 RUNNING
7440- AD 0D A0 3600 WAIT.ACK LDA VIA1IFR   ;GET INTERRUPT FLAGS
7443- 29 20          3605      AND #$20    ;CHECK FOR T2 TIMEOUT
7445- D0 0B          3610      BNE TIME.OUT ;...TIMED OUT BEFORE ACK
7447- 20 49 75 3615      JSR SRL.IC     ;CHK FOR ACK
744A- 4A              3620      LSR A
744B- B0 F3          3625      BCS WAIT.ACK ;...WAIT FOR ACK
744D- 58              3630      CLI        ;ALLOW INTERRUPTS AGAIN
744E- 60              3635      RTS        ;AND RETURN
                          3640 ;
                          3645 ;
                          3650 ;
                          3655 ;
                          3660 ; (SAME AS C-64 $EDAD/VIC 20 $EEB4)
744F- A9 80          3665 DEV.NP LDA #$80    ;SET ST FOR DEVICE NOT PRESENT

```

```

7451- 2C      3670      .BY $2C
              3675 ;
              3680 ;
              3685 ;          SERIAL TIME OUT
              3690 ;
              3695 ; (SAME AS C-64 $EDB0/VIC 20 $EEB7)
7452- A9 01    3700 TIME.OUT LDA #$01      ;SET SEND TIMEOUT FLAG
7454- 20 6A 75 3705 T.OUT   JSR SET.ST    ;STORE TO STATUS
7457- 58      3710      CLI              ;ALLOW INTERRUPTS AGAIN
7458- 18      3715      CLC
7459- 90 4F    3720      BCC UNLST1      ;...CLR ATN, ETC.
              3725 ;
              3730 ;
              3735 ;          SEND LISTEN SA
              3740 ;
              3745 ; (SAME AS C-64 $EDB9/VIC 20 $EEC0)
745B- 8D 53 01 3750 SECOND STA DF.CHR
745E- 20 D9 73 3755      JSR BUS.DIR
              3760 ;
              3765 ;
              3770 ;          CLEAR ATN
              3775 ;
              3780 ; (SAME AS C-64 $EDBE/VIC 20 $EEC5)
7461- AD 0F A0 3785 ATN.CLR LDA VIA1DRA
7464- 29 F7    3790      AND #$F7      ;RELEASE ATN
7466- 8D 0F A0 3795      STA VIA1DRA
7469- 60      3800      RTS
              3805 ;
              3810 ;
              3815 ;          SEND TALK SA
              3820 ;
              3825 ; (SAME AS C-64 $EDC7/VIC 20 $EECE)
746A- 8D 53 01 3830 TALK.SA STA DF.CHR
746D- 20 D9 73 3835      JSR BUS.DIR
7470- 78      3840      SEI              ;STOP INTERRUPTS
7471- 20 40 75 3845      JSR SRL.0
7474- 20 61 74 3850      JSR ATN.CLR    ;CLEAR ATN
7477- 20 25 75 3855      JSR CLK.ON
747A- 20 49 75 3860 T.SA   JSR SRL.IC   ;WAIT FOR DEVICE TO GRAB CLOCK
747D- 80 FB    3865      BCS T.SA
747F- 58      3870      CLI
7480- 60      3875      RTS
              3880 ;
              3885 ;
              3890 ;          SEND SERIAL DEFERRED
              3895 ;
              3900 ; (SAME AS C-64 $EDDD/VIC 20 $EEE4)
              3905 ; (SEND BYTE WITH HANDSHAKE)
7481- 2C 2E A6 3910 C1OUT  BIT DF.FLG   ;ANY PENDING?
7484- 30 06    3915      BMI CI1       ;...GOT ONE PENDING
7486- 38      3920      SEC
7487- 6E 2E A6 3925      ROR DF.FLG   ;SET FLAG FOR CHR PRESENT
748A- D0 05    3930      BNE CI2       ;...UNCONDITIONAL
748C- 48      3935 CI1    PHA          ;SAVE THIS CHR
748D- 20 E3 73 3940      JSR SEND.OUT  ;OUTPUT PREV. ONE
7490- 68      3945      PLA
7491- 8D 53 01 3950 CI2    STA DF.CHR   ;TEMP. STORED DEFERRED
7494- 18      3955      CLC

```

```

7495- 60      3960      RTS
              3965 ;
              3970 ;
              3975 ;          =====
              3980 ;          SEND 'UNTALK'
              3985 ;          =====
              3985 ; (SAME AS C-64 $EDEF/VIC 20 $EEF6)
7496- 78      3990 UNTALK SEI
7497- 20 2E 75 3995      JSR CLK.OFF
749A- AD 0F A0 4000      LDA VIA1DRA
749D- 09 08      4005      ORA #$08      ;TURN ON ATN
749F- 8D 0F A0 4010      STA VIA1DRA
74A2- A9 5F      4015      LDA #$5F
74A4- 2C      4020      .BY $2C
              4025 ;
              4030 ;
              4035 ;          =====
              4040 ;          SEND 'UNLISTEN'
              4045 ;          =====
              4045 ; (SAME AS C-64 $EDFE/VIC 20 $EF04)
74A5- A9 3F      4050 UNLISTEN LDA #$3F
74A7- 20 B1 73 4055      JSR CONTROL ;SEND LAST CHAR
74AA- 20 61 74 4060 UNLST1 JSR ATN.CLR ;CLEAR ATN
              4065 ;
74AD- 8A      4070 UNLST2 TXA      ;SAVE 'X'
74AE- A2 08      4075      LDX #$08      ;TIMING, C-64=0A,VIC 20=0B
74B0- CA      4080 UNLST3 DEX
74B1- D0 FD      4085      BNE UNLST3
74B3- AA      4090      TAX      ;RESTORE X
74B4- 20 25 75 4095      JSR CLK.ON ;ENABLE CLOCK IN
74B7- 4C 37 75 4100      JMP SRL.1 ;ENABLE DATA IN
              4105 ;
              4110 ;
              4115 ;          =====
              4120 ;          RECEIVE FROM SERIAL BUS
              4125 ;          =====
              4125 ; (SAVE AS C-64 $EE13/VIC 20 $EF19)
74BA- 78      4130 ACPTR SEI      ;STOP TALKING
74BB- A9 00      4135      LDA #0
74BD- 85 F9      4140      STA $BIT.CNT ;ONLY 1 EOI ALLOWED
74BF- 20 25 75 4145      JSR CLK.ON ;ENABLE CLOCK IN
74C2- 20 49 75 4150 RCV.CLK JSR SRL.IC ;GET INPUT CLOCK
74C5- 90 FB      4155      BCC RCV.CLK ;...WAIT FOR START
74C7- 20 37 75 4160      JSR SRL.1 ;INDICATE READY TO RECEIVE
74CA- A9 01      4165 RCV.T256 LDA #$01 ;TIMING MODE
74CC- 8D 09 A0 4170      STA VIA1T2H ;200 USEC DELAY
74CF- AD 0D A0 4175 RCV.TM1 LDA VIA1IFR ;CHK TIMER
74D2- 29 20      4180      AND #$20
74D4- D0 07      4185      BNE RCV.TM2 ;...TIMEOUT, CHK IF EOI
74D6- 20 49 75 4190      JSR SRL.IC ;GET INPUT CLOCK
74D9- B0 F4      4195      BCS RCV.TM1 ;...NO CLOCK YET
74DB- 90 18      4200      BCC RCV.BIT ;...INPUT CLOCK STARTED
74DD- A5 F9      4205 RCV.TM2 LDA $BIT.CNT ;BEEN HERE BEFORE?
74DF- F0 05      4210      BEQ RCV.EOI ;...NO, THIS MUST BE AN EOI
74E1- A9 02      4215      LDA #$02 ;ERROR, READ TIME OUT
74E3- 4C 54 74 4220      JMP T.OUT ;STORE AS STATUS
              4225 ;
74E6- 20 40 75 4230 RCV.EOI JSR SRL.0 ;ACK EOI
74E9- 20 AD 74 4235      JSR UNLST2 ;DELAY FOR 60 USEC
74EC- A9 40      4240      LDA #$40 ;SET ST EOI FLAG
74EE- 20 6A 75 4245      JSR SET.ST

```

```

74F1- E6 F9      4250      INC *BIT.CNT          ;ALLOW ONLY 1 EOI
74F3- D0 D5      4255      BNE RCV.T256          ;...UNCONDITIONAL
74F5- A9 08      4260 RCV.BIT LDA #8          ;SETUP FOR 8 BITS OF DATA
74F7- B5 F9      4265      STA *BIT.CNT
74F9- AD 0F A0    4270 RCV.TM3 LDA VIA1DRA
74FC- CD 0F A0    4275      CMP VIA1DRA
74FF- D0 F8      4280      BNE RCV.TM3          ;...NOT CHANGING
7501- 4A         4285      LSR A
7502- 90 F5      4290      BCC RCV.TM3          ;...WAIT DATA PRESENT CLOCK
7504- 4A         4295      LSR A
7505- 66 F8      4300      ROR *DISK.CHR          ;INTO CHR BUILD
7507- AD 0F A0    4305 RCV.TM4 LDA VIA1DRA
750A- CD 0F A0    4310      CMP VIA1DRA
750D- D0 F8      4315      BNE RCV.TM4          ;...NOT CHANGING
750F- 4A         4320      LSR A
7510- B0 F5      4325      BCS RCV.TM4          ;...WAIT CLOCK TO FALL
7512- C6 F9      4330      DEC *BIT.CNT          ;COUNT BIT IN
7514- D0 E3      4335      BNE RCV.TM3          ;...MORE BITS TO GET
7516- 20 40 75   4340      JSR SRL.0          ;FORCE DATA LOW
7519- A5 FC      4345      LDA *STATUS          ;CHK FOR EOI OR TIMEOUT
751B- F0 03      4350      BEQ =+4          ;...NONE
751D- 20 AD 74   4355      JSR UNLST2          ;DELAY A BIT & CLEAR DATA
7520- A5 F8      4360      LDA *DISK.CHR          ;LOAD CHR
7522- 58         4365      CLI
7523- 18         4370      CLC
7524- 60         4375      RTS
                        4380 ;
                        4385 ;
                        4390 ;          =====
                        4395 ;          SET CLOCK LINE HIGH (ie ON)
                        4400 ;          =====
                        4405 ; (SAME AS C-64 $EE85/VIC 20 $EF84)
7525- AD 0F A0    4405 CLK.ON  LDA VIA1DRA
7528- 29 EF      4410      AND #$EF          ;CLOCK OUT HIGH
752A- 8D 0F A0    4415      STA VIA1DRA
752D- 60         4420      RTS
                        4425 ;
                        4430 ;
                        4435 ;          =====
                        4440 ;          SET CLOCK LINE LOW (ie OFF)
                        4445 ;          =====
                        4450 ; (SAME AS C-64 $EE8E/VIC 20 $EF8D)
752E- AD 0F A0    4450 CLK.OFF LDA VIA1DRA
7531- 09 10      4455      ORA #$10          ;CLOCK OUT LOW
7533- 8D 0F A0    4460      STA VIA1DRA
7536- 60         4465      RTS
                        4470 ;
                        4475 ;
                        4480 ;          =====
                        4485 ;          SERIAL OUTPUT '1'
                        4490 ;          =====
                        4495 ; (SAME AS C-64 $EE97/VIC 20 $E4A0)
7537- AD 0F A0    4495 SRL.1  LDA VIA1DRA ;GET CONTROL REG
753A- 29 DF      4500      AND #$DF          ;SET DATA OUT HIGH
753C- 8D 0F A0    4505      STA VIA1DRA
753F- 60         4510      RTS
                        4515 ;
                        4520 ;
                        4525 ;          =====
                        4530 ;          SERIAL OUTPUT '0'
                        4535 ;          =====
                        4535 ; (SAME AS C-64 $EEA0/VIC 20 $E4A9)

```



```

7540- AD 0F A0 4540 SRL.0      LDA VIA1DRA ;GET CONTROL REG
7543- 09 20 4545              ORA #020 ;SET DATA OUT LOW
7545- 8D 0F A0 4550              STA VIA1DRA
7548- 60 4555              RTS
      4560 ;
      4565 ;
      4570 ;
      4575 ;
      4580 ; (SAME AS C-64 $EEA9/VIC 20 $E4B2)
7549- AD 0F A0 4585 SRL.IC     LDA VIA1DRA ;GET CURRENT INPUT
754C- CD 0F A0 4590              CMP VIA1DRA ;MAKE SURE NOT CHANGING
754F- D0 F8 4595              BNE SRL.IC ;...CHANGED, TRY AGAIN
7551- 4A 4600              LSR A ;CLOCK TO CARRY
7552- 60 4605              RTS
      4610 ;
      4615 ;
      4620 ;
      4625 ;
      4630 ; (SAME AS C-64 $EEBE/VIC 20 $EF96)
7553- A9 04 4635 DLY.1MS     LDA #4 ;*** TIMING ***
7555- 8D 09 A0 4640              STA VIA1T2H
7558- AD 0D A0 4645 DLY1     LDA VIA1IFR
755B- 29 20 4650              AND #020
755D- F0 F9 4655              BEQ DLY1
755F- 60 4660              RTS
      4665 ;
      4670 ;
      4675 ;
      4680 ;
      4685 ; (SAME AS C-64 $F0A4/VIC 20 $F160)
7560- 48 4690 S.CHK.IDL     PHA ;SAVE 'A'
7561- AD 0F A0 4695 C.IDL     LDA VIA1DRA
7564- 29 04 4700              AND #04
7566- F0 F9 4705              BEQ C.IDL ;...WAIT FOR ATN TO CLEAR
7568- 68 4710              PLA
7569- 60 4715              RTS
      4720 ;
      4725 ;
      4730 ;
      4735 ;
      4740 ; (SAME AS C-64 $FE1C/VIC 20 $FE6A)
756A- 05 FC 4745 SET.ST     ORA *STATUS
756C- 85 FC 4750              STA *STATUS
756E- 38 4755              SEC
756F- 60 4760              RTS
      4765 ;
7570- 0D 0A 4770 MESSAGE    .BY $0D $0A
7572- 3D 3D 3D 4775          .BY '=====
7575- 3D 3D 3D
7578- 3D 3D 3D
757B- 3D 3D 3D
757E- 3D 3D 3D
7581- 3D 3D 3D
7584- 3D 3D 3D
7587- 3D 3D 3D
758A- 3D 3D 3D
758D- 3D 3D 3D
7590- 3D 3D 3D

```

7593- 0D 0A	4780	.BY \$0D \$0A
7595- 53 59 4D	4785	.BY 'SYM-1 1541 DISK OPERATING SYSTEM'
7598- 2D 31 20		
7598- 31 35 34		
759E- 31 20 44		
75A1- 49 53 4B		
75A4- 20 4F 50		
75A7- 45 52 41		
75AA- 54 49 4E		
75AD- 47 20 53		
75B0- 59 53 54		
75B3- 45 4D		
75B5- 0D 0A	4790	.BY \$0D \$0A
75B7- 43 6F 70	4795	.BY 'Copyright 1984 by Ronald A. Jordan'
75BA- 79 72 69		
75BD- 67 68 74		
75C0- 20 31 39		
75C3- 38 34 20		
75C6- 62 79 20		
75C9- 52 6F 6E		
75CC- 61 6C 64		
75CF- 20 41 2E		
75D2- 20 4A 6F		
75D5- 72 64 61		
75D8- 6E		
75D9- 0D 0A	4800	.BY \$0D \$0A
75DB- 3D 3D 3D	4805	.BY '====='
75DE- 3D 3D 3D		
75E1- 3D 3D 3D		
75E4- 3D 3D 3D		
75E7- 3D 3D 3D		
75EA- 3D 3D 3D		
75ED- 3D 3D 3D		
75F0- 3D 3D 3D		
75F3- 3D 3D 3D		
75F6- 3D 3D 3D		
75F9- 3D 3D 3D		
75FC- 0D 0A 0A	4810	.BY \$0D \$0A \$0A \$00
75FF- 00		
	4815 ;	
	4820 SYM.END	.CT RAE1541 0



```

0001      .EJ
5000 ; RAE INTERFACE ROUTINES FOR THE COMMODORE 1541 DISK DRIVE
5005 ; INTERFACED VIA RAE GET, PUT AND DC VECTORS
5010 ; (MODIFIED FROM NICK VRTIS'S RAE-1541 ROUTINES)
5015 ;
5020 ;      BY RONALD A. JORDAN - FEB. 1984
5025 ;
5030      .BA $7800
5035 ;
5040 PURECL      .DE $C8      ;RELOCATING BUFFER ADDR LO
5045 TPRES      .DE $D3      ;PRESENT END OF TEXT FILE
5050 DISKC.VEC   .DE $EC      ;DISK COMMAND VECTOR
5055 DISKI      .DE $EE      ;1=DISK INPUT, ELSE TAPE
5060 DISKO      .DE $EF      ;1=DISK OUTPUT, ELSE TAPE
5065 DISKO.VEC   .DE $F4      ;PUT VECTOR
5070 DISKI.VEC   .DE $F6      ;GET VECTOR
5075 ;
5080 TXST      .DE $100      ;START OF TEXT FILE
5085 TXEN      .DE $102      ;END OF TEXT FILE
5090 STST      .DE $104      ;START OF LABEL FILE
5095 STEN      .DE $106      ;END OF LABEL FILE
5100 FIRST      .DE $108      ;FIRST LINE #
5105 FILE.NO    .DE $110      ;CURRENT FILE NUMBER
5110 HEX/DEC    .DE $111      ;0=HEX, 1=DEC
5115 CUR.SAVE   .DE $11C      ;USED TO LOCATE LAST LINE
5120 TSTART     .DE $124      ;TAPE START ADDRESS
5125 TEND       .DE $126      ;TAPE END ADDRESS
5130 HFILE/NO    .DE $128      ;HEADER FILE NNUMBER
5135 HSTART     .DE $129      ;HEADER START ADDRESS
5140 HEND       .DE $12B      ;HEADER END ADDRESS
5145 CRT.BUF    .DE $135      ;CRT BUFFER
5150 ;
5155 ; RAE ADDRESSES
5160 RAE.WARM    .DE $B003      ;RAE WARM START
5165 ERROR      .DE $B00E      ;RAE ERROR VECTOR
5170 LINE.ADR    .DE $B214      ;FIND ADDR OF LINE
5175 TAPE1.OFF   .DE $E318      ;TURN OFF READ TAPE
5180 TAPE0.OFF   .DE $E30F      ;TURN OFF WRITE TAPE
5185 SET.EOT     .DE $E597      ;PUT EOT AFTER TAPE LOAD
5190 TAPE.GET    .DE $EF68      ;TAPE RE-ENTRY AFTER GET
5195 TAPE.PUT    .DE $EF95      ;TAPE RE-ENTRY AFTER PUT
5200 ;
5205 ; RAE INITIALIZATION PARAMETERS
5210 TXTLADR      .DE $200      ;START OF TEXT FILE
5215 TXTHADR      .DE $5FFC      ;END OF TEXT FILE
5220 LBLADR      .DE $6000      ;START OF LABEL FILE
5225 LBLHADR      .DE $6EFC      ;END OF LABEL FILE
5230 BUFADR      .DE $6F00      ;BUFFER AREA
5235 ;
5240 ;      =====
5245 ;      RAE COLD START
5250 ;      =====
7800- 20 86 BB 5255 RAE.INIT JSR ACCESS
7803- A9 17 5260 LDA #L,RAE.COLD ;RAE START ADR TO P3
7805- 8D 4A A6 5265 STA P3
7808- A9 7A 5270 LDA #H,RAE.COLD
780A- 8D 4B A6 5275 STA P3+1
780D- 4C 55 88 5280 JMP EXECUTE ;DO MONITOR EXECUTE

```

```

5285 ;
5290 ;
5295 ;
5300 ;
=====
7810- A9 01 5305 RAE.SETUP LDA #1 ;ENABLE GET & PUT I/O
7812- 85 EE 5310 STA *DISKI
7814- 85 EF 5315 STA *DISKO
7816- A9 66 5320 LDA #L,GET.VEC ;SETUP RAE VECTORS
7818- 85 F6 5325 STA *DISKI.VEC
781A- A9 78 5330 LDA #H,GET.VEC
781C- 85 F7 5335 STA *DISKI.VEC+1
781E- A9 B3 5340 LDA #L,DISK.RAE
7820- 85 EC 5345 STA *DISKC.VEC
7822- A9 79 5350 LDA #H,DISK.RAE
7824- 85 ED 5355 STA *DISKC.VEC+1
7826- A9 60 5360 LDA #*60 ;SET RETURN IN NEW DC VEC
7828- 8D 10 A6 5365 STA NEW.DCV
782B- A9 71 5370 LDA #L,PUT.VEC
782D- 85 F4 5375 STA *DISKO.VEC
782F- A9 79 5380 LDA #H,PUT.VEC
7831- 85 F5 5385 STA *DISKO.VEC+1
7833- A9 00 5390 LDA #L,BUFADR ;SETUP RAE FOR 28K
7835- 85 C8 5395 STA *PURECL
7837- A9 6F 5400 LDA #H,BUFADR
7839- 85 C9 5405 STA *PURECL+1
783B- A9 00 5410 LDA #L,TXTLADR
783D- 8D 00 01 5415 STA TXST
7840- A9 02 5420 LDA #H,TXTLADR
7842- 8D 01 01 5425 STA TXST+1
7845- A9 FC 5430 LDA #L,TXTHADR
7847- 8D 02 01 5435 STA TXEN
784A- A9 5F 5440 LDA #H,TXTHADR
784C- 8D 03 01 5445 STA TXEN+1
784F- A9 00 5450 LDA #L,LBLLADR
7851- 8D 04 01 5455 STA STST
7854- A9 60 5460 LDA #H,LBLLADR
7856- 8D 05 01 5465 STA STST+1
7859- A9 FC 5470 LDA #L,LBLHADR
785B- 8D 06 01 5475 STA STEN
785E- A9 6E 5480 LDA #H,LBLHADR
7860- 8D 07 01 5485 STA STEN+1
5490 ;
7863- 4C 03 B0 5495 JMP RAE.WARM ;WARM START RAE
5500 ;
5505 ;
5510 ;
5515 ;
=====
7866- AD 47 A6 5520 GET.VEC LDA DISK.DEV ;DISK OR TAPE?
7869- D0 0B 5525 BNE DISK.GET ;...GO TO DISK
5530 ;
786B- AD 11 01 5535 LDA HEX/DEC ;STD TAPE .CT PATCH
786E- D0 03 5540 BNE =+4
7870- 8D 10 01 5545 STA FILE.NO ;FILE NUMBER =0
7873- 4C 68 EF 5550 JMP TAPE.GET
5555 ;
7876- C9 08 5560 DISK.GET CMP #8 ;CHECK DEVICE #
7878- 90 6D 5565 BCC TO.ERR32 ;...<8, ERROR
787A- 20 18 E3 5570 JSR TAPE1.OFF ;RECORD TAPE OFF

```

```

787D- AD 25 01 5575      LDA TSTART+1      ;HEADER OR DATA PASS?
7880- C9 01 5580      CMP #01              ;HEADER IF PAGE 1
7882- D0 15 5585      BNE CHK.GTYPE        ;...NO, DO .CT OR LOAD
                    5590 ;
                    5595 ;HEADER FOR 'GET'
                    5600 ;=====
7884- A9 00 5605      LDA #0                ;ZERO HEADER INFO
7886- 8D 28 01 5610      STA HFILE.NO
7889- 8D 10 01 5615      STA FILE.NO
788C- 8D 29 01 5620      STA HSTART
788F- 8D 2A 01 5625      STA HSTART+1
7892- 8D 2B 01 5630      STA HEND
7895- 8D 2C 01 5635      STA HEND+1
7898- 60 5640      RTS
                    5645 ;
                    5650 ;CHECK FOR DISK .CT OR LOAD AND FINISH
                    5655 ;=====
7899- AD 11 01 5660      CHK.GTYPE LDA HEX/DEC      ;.CT?
789C- F0 4C 5665      BEQ DISK.CT          ;...YES, DISK .CT
789E- A2 00 5670      LDX #0                ;NAME TO START
78A0- 20 2C 79 5675      JSR GET.NAM        ;NAME TO FNAME
78A3- AD 47 A6 5680      DISK.GET1 LDA DISK.DEV    ;SET DEVICE #
78A6- 8D 4E A6 5685      STA C.DEV
78A9- 8D 4A A6 5690      STA P3              ;NON-0 FOR RELOCATED LOAD
78AC- AD 24 01 5695      LDA TSTART          ;START ADR TO P2
78AF- 8D 4C A6 5700      STA P2
78B2- AD 25 01 5705      LDA TSTART+1
78B5- 8D 4D A6 5710      STA P2+1
78B8- 20 03 70 5715      JSR LOADV          ;DO LOAD
78BB- B0 1D 5720      BCS ERR33             ;...LOAD ERROR
78BD- A5 FE 5725      LDA *CURAD            ;SET NEW EOT
78BF- 8D 26 01 5730      STA TEND
78C2- A5 FF 5735      LDA *CURAD+1
78C4- 8D 27 01 5740      STA TEND+1
78C7- CD 03 01 5745      CMP TXEN+1         ;CHK FOR TEXT OVERFLOW
78CA- D0 05 5750      BNE =+6
78CC- A5 FE 5755      LDA *CURAD
78CE- CD 02 01 5760      CMP TXEN
78D1- B0 0F 5765      BCS ERR31
78D3- A2 00 5770      LDX #0
78D5- 20 97 E5 5775      JSR SET.EOT        ;END ADR TO TPRES & STORE EOT
78D8- 8A 5780      TXA
78D9- 60 5785      RTS
                    5790 ;
78DA- 20 71 81 5795      ERR33 JSR ERMSG        ;DISPLAY STATUS
78DD- A2 33 5800      LDX #33              ;DISK I/O ERROR
78DF- 6C 0E B0 5805      JMP (ERROR)
                    5810 ;
78E2- A2 31 5815      ERR31 LDX #31          ;TEXT FILE OVERFLOW
78E4- 6C 0E B0 5820      JMP (ERROR)
                    5825 ;
78E7- 4C 27 79 5830      TO.ERR32 JMP ERR32
                    5835 ;
                    5840 ;CONTINUE ON DISK
                    5845 ;=====
78EA- A0 02 5850      DISK.CT LDY #2          ;POINT PAST LINE #
78EC- A9 3B 5855      LDA #';              ;MAKE 1ST LINE A VALID LINE
78EE- 91 D3 5860      STA (TPRES),Y

```

78F0-	A9	FF	5865		LDA #\$FF
78F2-	8D	09 01	5870		STA FIRST+1 ;SEARCH FOR HIGHEST #
78F5-	20	14 B2	5875		JSR LINE.ADR
78F8-	AD	1C 01	5880		LDA CUR.SAVE
78FB-	85	FE	5885		STA *CURAD
78FD-	AD	1D 01	5890		LDA CUR.SAVE+1
7900-	85	FF	5895		STA *CURAD+1
7902-	A0	01	5900		LDY #1 ;JUST BEFORE 1ST CHR
7904-	C8		5905	FND.CT	INY ;NEXT CHR
7905-	B1	FE	5910		LDA (CURAD),Y
7907-	30	19	5915		BMI ERR34 ;INVALID FNAME
7909-	C9	2E	5920		CMP #' ;LOOK FOR '.'
790B-	D0	F7	5925		BNE FND.CT ;...KEEP LOOKING
790D-	C8		5930		INY
790E-	B1	FE	5935		LDA (CURAD),Y
7910-	C9	43	5940		CMP #'C ;.C?
7912-	D0	F0	5945		BNE FND.CT ;...KEEP LOOKING
7914-	C8		5950		INY
7915-	B1	FE	5955		LDA (CURAD),Y
7917-	C9	54	5960		CMP #'T ;.CT?
7919-	D0	E9	5965		BNE FND.CT ;...KEEP LOOKING
791B-	C8		5970		INY
791C-	20	46 79	5975		JSR GN.MOVE ;GET NAME FROM .CT
791F-	4C	A3 78	5980		JMP DISK.GET1 ;FINISH WITH LOAD
			5985 ;		
7922-	A2	34	5990	ERR34	LDX #\$34 ;INVALID FNAME
7924-	6C	0E B0	5995		JMP (ERROR)
7927-	A2	32	6000	ERR32	LDX #\$32 ;INVALID DEVICE #
7929-	6C	0E B0	6005		JMP (ERROR)
			6010 ;		
			6015 ;		
			6020 ;		
			6025 ;		
792C-	A0	02	6030	GET.NAM	LDY #2 ;AT LEAST PU/ OR GE/
792E-	20	86 8B	6035	G.NAM	JSR ACCESS
7931-	A9	35	6040		LDA #L,CRT.BUF
7933-	85	FE	6045		STA *CURAD
7935-	A9	01	6050		LDA #H,CRT.BUF
7937-	85	FF	6055		STA *CURAD+1
7939-	B1	FE	6060	GN.DELIM	LDA (CURAD),Y ;GET FROM BUFFER
793B-	C9	2F	6065		CMP #' / ;FIND FNAME START
793D-	F0	15	6070		BEQ GNM.LP ;...YES, NAME STARTS NEXT
793F-	C8		6075		INY ;ELSE NEXT CHR
7940-	C0	50	6080		CPY #80 ;END OF BUFFER?
7942-	F0	DE	6085		BEQ ERR34 ;NAME/COMMAND ERROR
7944-	D0	F3	6090		BNE GN.DELIM ;...ALWAYS
7946-	A2	00	6095	GN.MOVE	LDX #0 ;NAME TO START OF FNAME
7948-	20	86 8B	6100		JSR ACCESS
794B-	B1	FE	6105	GN.MOV1	LDA (CURAD),Y
794D-	C9	20	6110		CMP #\$20 ;SKIP LEADING BLANKS
794F-	D0	04	6115		BNE GNM.LP1 ;...1ST NON-BLANK
7951-	C8		6120		INY
7952-	D0	F7	6125		BNE GN.MOV1 ;...ALWAYS
7954-	C8		6130	GNM.LP	INY
7955-	B1	FE	6135	GNM.LP1	LDA (CURAD),Y
7957-	29	7F	6140		AND #\$7F ;STRIP EOL BIT
7959-	C9	20	6145		CMP #\$20 ;BLANKS?
795B-	F0	10	6150		BEQ GN.END ;...YES, END

```

795D- C9 09      6155      CMP #09      ;TAB?
795F- F0 0C      6160      BEQ GN.END    ;...YES, END
7961- 9D 35 01    6165      STA FNAME,X    ;ELSE SAVE CHR
7964- E8          6170      INX
7965- E0 1E      6175      CPX #30      ;NAME TOO LONG?
7967- F0 B9      6180      BEQ ERR34      ;INVALID NAME
7969- B1 FE      6185      LDA (CURAD),Y    ;CHK FOR EOL
796B- 10 E7      6190      BPL GNM.LP     ;...NOT YET
796D- 8E 54 01    6195      STX FN.LEN     ;SAVE NAME LENGTH
7970- 60          6200      RTS
                        6205 ;
                        6210 ;
                        6215 ;
                        6220 ;
                        6225      PUT.VEC
7971- AD 47 A6    6225      LDA DISK.DEV    ;OUTPUT TO DISK?
7974- D0 03      6230      BNE DISK.PUT
7976- 4C 95 EF    6235      JMP TAPE.PUT    ;ELSE NORMAL TAPE
                        6240 ;
7979- C9 08      6245      DISK.PUT      CMP #8      ;CHECK DRIVE #
797B- 90 AA      6250      BCC ERR32      ;...,<8, ERROR
797D- 20 0F E3    6255      JSR TAPE0.OFF
7980- AD 25 01    6260      LDA TSTART+1    ;HEADER OR DATA PASS?
7983- C9 01      6265      CMP #01
7985- D0 03      6270      BNE PUT.DATA    ;...NO, THEN DATA PASS
7987- A9 00      6275      LDA #0      ;SET OK FLAG
7989- 60          6280      RTS      ;AND RETURN THIS PASS
                        6285 ;
798A- A2 00      6290      PUT.DATA      LDX #0      ;FILE NAME TO BUFF START
798C- 20 2C 79    6295      JSR GET.NAM    ;NAME FROM CRT BUFFER
798F- AD 47 A6    6300      LDA DISK.DEV    ;SET DISK DEVICE #
7992- BD 4E A6    6305      STA C.DEV
7995- AD 29 01    6310      LDA HSTART      ;SETUP SAVE PARMS
7998- BD 4C A6    6315      STA P2
799B- AD 2A 01    6320      LDA HSTART+1
799E- BD 4D A6    6325      STA P2+1
79A1- AD 2B 01    6330      LDA HEND *      ;POINTS TO LAST BYTE
79A4- BD 4A A6    6335      STA P3
79A7- AD 2C 01    6340      LDA HEND+1
79AA- BD 4B A6    6345      STA P3+1
79AD- 20 06 70    6350      JSR SAVEV
79B0- B0 34      6355      BCS TO.ERR33    ;...DISK I/O ERROR
79B2- 60          6360      RTS
                        6365 ;
                        6370 ;
                        6375 ;
                        6380 ;
                        6385      DISK.RAE
79B3- C0 50      6385      CPY #00      ;ANY COMMAND GIVEN?
79B5- F0 2C      6390      BEQ TO.ERR34    ;...INVALID COMMAND
79B7- B9 35 01    6395      LDA CRT.BUF,Y
79BA- C9 23      6400      CMP #'*      ;DEVICE #?
79BC- F0 2E      6405      BEQ LOCAL.DEV    ;...YES, CHANGE #
79BE- C9 21      6410      CMP #'!      ;DISK STATUS?
79C0- F0 37      6415      BEQ PRT.ST      ;...YES
79C2- C9 3F      6420      CMP #'?      ;DIRECTORY?
79C4- F0 41      6425      BEQ DIR.LIST    ;...YES
79C6- C9 2F      6430      CMP #'/'      ;DISK COMMAND?
79C8- F0 03      6435      BEQ DISK.CD     ;...YES
79CA- 4C 10 A6    6440      JMP NEW.DCV    ;RETURN OR ADD MORE COMMANDS

```

```

6445 ;
6450 ;SEND 1541 DISK COMMAND
6455 ;=====
79CD- AD 47 A6 6460 DISK.CD LDA DISK.DEV ;GET DISK DEVICE #
79D0- C9 08 6465 CMP #8
79D2- 90 15 6470 BCC TOO.ERR32 ;...<8, ERROR
79D4- 18 6475 CLC
79D5- 8D 4E A6 6480 STA C.DEV
79D8- A2 00 6485 LDX #0 ;PUT DISK COMMAND AT START
79DA- 20 2E 79 6490 JSR G.NAM
79DD- 20 09 70 6495 JSR DISKCMDV ;USE EXTENDED COMMAND
79E0- B0 04 6500 BCS TOO.ERR32 ;...ERROR
79E2- 60 6505 RTS
6510 ;
6515 ;ERRORS
6520 ;=====
79E3- 4C 22 79 6525 TO.ERR34 JMP ERR34
79E6- 4C DA 78 6530 TO.ERR33 JMP ERR33
79E9- 4C 27 79 6535 TOO.ERR32 JMP ERR32
6540 ;
6545 ;CHANGE CURRENT DEVICE NUMBER
6550 ;=====
79EC- CB 6555 LOCAL.DEV INY
79ED- B9 35 01 6560 LDA CRT.BUF,Y ;GET DRIVE #
79F0- 20 75 82 6565 JSR ASCNIB ;CONVERT TO HEX
79F3- B0 F4 6570 BCS TOO.ERR32 ;...NON HEX CHR
79F5- 8D 47 A6 6575 STA DISK.DEV
79F8- 60 6580 RTS
6585 ;
6590 ;DISPLAY DISK STATUS
6595 ;=====
79F9- AD 47 A6 6600 PRT.ST LDA DISK.DEV ;GET DEVICE #
79FC- C9 08 6605 CMP #8
79FE- 90 E9 6610 BCC TOO.ERR32 ;...<8, ERROR
7A00- 18 6615 CLC
7A01- 8D 4E A6 6620 STA C.DEV
7A04- 4C 0C 70 6625 JMP DISKSTV ;USE EXTENDED COMMAND
6630 ;
6635 ;DISPLAY DISK DIRECTORY
6640 ;=====
7A07- A2 01 6645 DIR.LIST LDX #1 ;SET LENGTH OF NAME
7A09- AD 47 A6 6650 LDA DISK.DEV
7A0C- C9 08 6655 CMP #8
7A0E- 90 D9 6660 BCC TOO.ERR32 ;...<8, ERROR
7A10- 18 6665 CLC
7A11- 8D 4E A6 6670 STA C.DEV
7A14- 4C 0F 70 6675 JMP DIRENTV ;USE EXTENDED COMMAND
6680 ;
7A17- 47 42 30 6685 RAE.COLD .BY 'GB000' $0D
7A1A- 30 30 0D
7A1D- 52 55 20 6690 .BY 'RU $A600' $0D
7A20- 24 41 36
7A23- 30 30 0D
7A26- 00 6695 .BY 0
6700 ;
6705 RAE.END .CT BAS1541 0

```

```

0001 .EJ
7000 ; BASIC INTERFACE ROUTINES FOR THE COMMODORE 1541 DISK
7005 ; DRIVE. INTERFACED THROUGH INVEC AND OUTVEC
7010 ;
7015 ; BY RONALD A. JORDAN - MARCH 1984
7020 ;
7025 ;
7030 BUFF .DE $1E ;BUFFER POINTER
7035 PSAD .DE $7B ;START SOURCE TEXT
7040 VSAD .DE $7D ;START SIMPLE VARIABLES
7045 HIMEM .DE $87 ;TOP OF BASIC RAM
7050 CHRGET .DE $CC ;GET NEXT CHARACTER
7055 CHRGOT .DE $D2 ;GET LAST CHARACTER
7060 TXTPTR .DE TPRES ;TEXT POINTER, LOW
7065 ACC .DE $ED ;SAVE ACC
7070 XREG .DE DISKI ;SAVE X
7075 YREG .DE DISKO ;SAVE Y
7080 ;
7085 ; BASIC ROUTINES
7090 BASWARM .DE $C27E ;BASIC WARM START
7095 SCRATCH .DE $C458
7100 RUN.CMD .DE $C707
7105 ;
7110 ; =====
7115 ; BASIC COLD START
7120 ; =====
7A27- 20 86 8B 7125 BAS.INIT JSR ACCESS
7A2A- A9 4E 7130 LDA #L,BAS.START
7A2C- 8D 4A A6 7135 STA P3
7A2F- A9 7C 7140 LDA #H,BAS.START
7A31- 8D 4B A6 7145 STA P3+1
7A34- 4C 55 8B 7150 JMP EXECUTE
7155 ;
7A37- 20 58 C4 7160 BAS.COLD JSR SCRATCH ;NEW AND CLEAR BASIC TEXT
7A3A- AD 3A A6 7165 LDA SCRA ;SET INVEC FOR EXECUTE RTS
7A3D- AC 3B A6 7170 LDY SCRA+1
7A40- 8D 0D A6 7175 STA NEW.INV+1
7A43- 8C 0E A6 7180 STY NEW.INV+2
7A46- A9 8A 7185 LDA #L,INPUT ;INPUT TO ... INVEC
7A48- A0 7A 7190 LDY #H,INPUT
7A4A- 8D 3A A6 7195 STA SCRA
7A4D- 8C 3B A6 7200 STY SCRA+1
7A50- 4C 6C 7A 7205 JMP WEDGE2
7210 ;
7A53- 20 86 8B 7215 WEDGE JSR ACCESS
7A56- AD 61 A6 7220 LDA INVEC+1 ;SETUP NEW INVEC
7A59- AC 62 A6 7225 LDY INVEC+2
7A5C- 8D 0D A6 7230 STA NEW.INV+1
7A5F- 8C 0E A6 7235 STY NEW.INV+2
7A62- A9 8A 7240 LDA #L,INPUT ;INVEC INTO INPUT
7A64- A0 7A 7245 LDY #H,INPUT
7A66- 8D 61 A6 7250 STA INVEC+1
7A69- 8C 62 A6 7255 STY INVEC+2
7260 ;
7A6C- A9 00 7265 WEDGE2 LDA #0
7A6E- 8D 53 A6 7270 STA TECH0 ;SET FOR NO ECHO
7A71- AD 64 A6 7275 LDA OUTVEC+1 ;SETUP NEW OUTVEC
7A74- AC 65 A6 7280 LDY OUTVEC+2

```



7A77-	8D 0A A6	7285	STA NEW.OUTV+1
7A7A-	8C 0B A6	7290	STY NEW.OUTV+2
7A7D-	A9 CA	7295	LDA #L, OUTPUT ;OUTVEC GOES TO OUTPUT
7A7F-	A0 7A	7300	LDY #H, OUTPUT
7A81-	8D 64 A6	7305	STA OUTVEC+1
7A84-	8C 65 A6	7310	STY OUTVEC+2
7A87-	4C 7E C2	7315	JMP BASWARM
		7320 ;	
		7325 ;	
		7330 ;	=====
		7335 ;	INVEC ENTRY POINT
			=====
7A8A-	20 0C A6	7340 INPUT	JSR NEW.INV ;GET CHARACTER
7A8D-	29 7F	7345	AND #7F ;STRIP BIT 7
7A8F-	C9 03	7350 CTROL	CMP #3 ;MONITOR JUMP?
7A91-	D0 26	7355	BNE SEND ;NO-
7A93-	20 86 8B	7360	JSR ACCESS
7A96-	AD 0A A6	7365	LDA NEW.OUTV+1 ;RESTORE I/O
7A99-	8D 64 A6	7370	STA OUTVEC+1
7A9C-	AD 0B A6	7375	LDA NEW.OUTV+2
7A9F-	8D 65 A6	7380	STA OUTVEC+2
7AA2-	AD 0D A6	7385	LDA NEW.INV+1 ;RESTORE I/O
7AA5-	8D 61 A6	7390	STA INVEC+1
7AA8-	AD 0E A6	7395	LDA NEW.INV+2
7AAB-	8D 62 A6	7400	STA INVEC+2
7AAE-	A9 80	7405	LDA #80 ;RESET ECHO FLG
7AB0-	8D 53 A6	7410	STA TECHO
7AB3-	20 35 80	7415	JSR USRENT
7AB6-	4C 53 7A	7420	JMP WEDGE ;RETURN BY "G CR"
		7425 ;	
7AB9-	4C 09 A6	7430 SEND	JMP NEW.OUTV
		7435 ;	
		7440 ;	
		7445 ;	=====
		7450 ;	SAVE ALL REGISTERS
			=====
7ABC-	85 ED	7455 SAVEREG	STA *ACC
7ABE-	86 EE	7460 SAVEXY	STX *XREG
7AC0-	84 EF	7465	STY *YREG
7AC2-	60	7470	RTS
		7475 ;	
		7480 ;	
		7485 ;	=====
		7490 ;	RESTORE ALL REGISTERS
			=====
7AC3-	A5 ED	7495 RESTORREG	LDA *ACC
7AC5-	A6 EE	7500 RESTORXY	LDX *XREG
7AC7-	A4 EF	7505	LDY *YREG
7AC9-	60	7510	RTS
		7515 ;	
		7520 ;	
		7525 ;	=====
		7530 ;	OUTVEC ENTRY POINT
			=====
7ACA-	20 BC 7A	7535 OUTPUT	JSR SAVEREG ;SAVE REGS
7ACD-	A2 05	7540	LDX #5
7ACF-	68	7545 PULL2	PLA ;PULL STACK
7AD0-	CA	7550	DEX ;TO ALLOW TOUT NOT OUTCHR
7AD1-	10 FC	7555	BPL PULL2
7AD3-	20 C3 7A	7560	JSR RESTORREG
		7565 ;	
7AD6-	C9 0D	7570	CMP #0D ;LOOK FOR CR

```

7AD8- D0 09      7575      BNE SEND2
7ADA- 20 D2 00    7580      JSR CHRGET      ;GET LAST CHR
7ADD- C9 23      7585      CMP #'#
7ADF- F0 08      7590      BEQ NEWCMDS      ;YES, CHECK NEW COMMANDS
7AE1- A9 0D      7595      LDA #0D          ;RESTORE CR
7AE3- 20 09 A6    7600 SEND2 JSR NEW.OUTV
7AE6- 4C C3 7A    7605      JMP RESTOREG
                        7610 ;
                        7615 ;
                        7620 ;          PULL STACK BACK TO COMMAND PROCESSOR
7AE9- A2 07      7625 NEWCMDS LDX #7
7AEB- 68          7630 PULL3   PLA
7AEC- CA          7635        DEX
7AED- 10 FC      7640        BPL PULL3
                        7645 ;
7AEF- A0 00      7650        LDY #0          ;SET INDEX FOR CMD POINTER
7AF1- A2 00      7655 CMD.CHK LDX #0
7AF3- B5 1E      7660 C.CHK   LDA *BUFF,X
7AF5- D9 3F 7C   7665        CMP CMD.TABLE,Y
7AF8- D0 0F      7670        BNE NEXT.CMD ;NO, TRY NEXT CMD
7AFA- E8          7675        INX          ;MOVE TO NEXT CHR
7AFB- C8          7680        INY
7AFC- E0 03      7685        CPX #3          ;ONLY 3 CHRS / CMD
7AFE- D0 F3      7690        BNE C.CHK
7B00- B9 40 7C   7695        LDA CMD.TABLE+1,Y ;HIGH ADDRESS
7B03- 48          7700        PHA          ;SET RETURN
7B04- B9 3F 7C   7705        LDA CMD.TABLE,Y  ;LOW ADDRESS
7B07- 48          7710        PHA
7B08- 60          7715        RTS          ;MATCH, GOOD RETURN
                        7720 ;
7B09- C8          7725 NEXT.CMD INY          ;NEXT CMD IN TABLE
7B0A- C8          7730        INY
7B0B- 18          7735        CLC
7B0C- C0 0F      7740        CPY #15        ;CHARS IN CMD TABLE
7B0E- 90 E1      7745        BCC CMD.CHK
7B10- 4C CC 00    7750 NXT.ERR JMP CHRGET ;SN ERR, MOVE TO NEXT CHR
                        7755 ;
                        7760 ;
                        7765 ;          =====
                        7770 ;          DISK LOAD & SAVE ROUTINES
                        7775 ;          =====
7B13- A9 01      7775 APPEND.PRG LDA #1
7B15- 2C          7780        .BY #2C
7B16- A9 00      7785 LOAD.PRG  LDA #0
7B18- 20 86 8B   7790        JSR ACCESS
7B1B- 8D 48 A6   7795        STA APPFL6
7B1E- AD 47 A6   7800        LDA DISK.DEV
7B21- C9 08      7805        CMP #8          ;CHECK DEVICE #
7B23- 90 68      7810        BCC D.ERR      ;...<8, ERROR
7B25- 8D 4E A6   7815        STA C.DEV
7B28- 8D 4A A6   7820        STA P3          ;NON ZERO, RELOCATE LOAD OPT.
7B2B- 20 15 7C   7825        JSR GET.FNAME
7B2E- B0 65      7830        BCS F.ERR      ;FILE NAME ERROR
7B30- AD 48 A6   7835        LDA APPFL6
7B33- F0 06      7840        BEQ L.PRG1     ;NON ZERO, APPEND LOAD
7B35- A6 7D      7845        LDX *VSAD      ;APPEND ADDRESS
7B37- A4 7E      7850        LDY *VSAD+1
7B39- D0 04      7855        BNE L.PRG2     ;ALWAYS
7B3B- A6 7B      7860 L.PRG1  LDX *PSAD      ;NORMAL ADDRESS

```

7B3D-	A4	7C	7865	LDY #PSAD+1
7B3F-	8E	4C A6	7870 L.PRG2	STX P2
7B42-	8C	4D A6	7875	STY P2+1
7B45-	20	03 70	7880	JSR LOADV
7B48-	B0	4F	7885	BCS L.ERR ;LOAD ERROR
7B4A-	A6	FE	7890	LDX #CURAD ;SET BASIC EOT
7B4C-	A4	FF	7895	LDY #CURAD+1
7B4E-	86	7D	7900	STX #VSAD
7B50-	84	7E	7905	STY #VSAD+1
7B52-	C4	88	7910	CPY #HIMEM+1 ;CHECK FOR TEXT OVERFLOW
7B54-	D0	02	7915	BNE L.PRG3
7B56-	E4	87	7920	CPX #HIMEM
7B58-	B0	47	7925 L.PRG3	BCS T.ERR ;TEXT OVERFLOW ERROR
7B5A-	60		7930	RTS
			7935 ;	
7B5B-	20	16 7B	7940 RUN.PRG	JSR LOAD.PRG
7B5E-	4C	07 C7	7945	JMP RUN.CMD
			7950 ;	
7B61-	20	86 8B	7955 SAVE.PRG	JSR ACCESS
7B64-	AD	47 A6	7960	LDA DISK.DEV
7B67-	C9	00	7965	CMP #8 ;CHECK DEVICE #
7B69-	90	22	7970	BCC D.ERR ;...<8, ERROR
7B6B-	8D	4E A6	7975	STA C.DEV
7B6E-	20	15 7C	7980	JSR GET.FNAME
7B71-	B0	22	7985	BCS F.ERR
7B73-	A6	7B	7990	LDX #PSAD
7B75-	A4	7C	7995	LDY #PSAD+1
7B77-	8E	4C A6	8000	STX P2
7B7A-	8C	4D A6	8005	STY P2+1
7B7D-	A6	7D	8010	LDX #VSAD
7B7F-	A4	7E	8015	LDY #VSAD+1
7B81-	8E	4A A6	8020	STX P3
7B84-	8C	4B A6	8025	STY P3+1
7B87-	20	06 70	8030	JSR SAVEV
7B8A-	B0	11	8035	BCS S.ERR ;SAVE ERROR
7B8C-	60		8040	RTS
			8045 ;	
			8050 ;	
			8055 ;	
			8060 ;	=====
			8065 ;	ERROR ROUTINES
				=====
7B8D-	A9	31	8070 D.ERR	LDA #DEV.ERR
7B8F-	D0	12	8075	BNE SEND.ERR
7B91-	A9	36	8080 DC.ERR	LDA #DCMD.ERR
7B93-	D0	0E	8085	BNE SEND.ERR
7B95-	A9	32	8090 F.ERR	LDA #FNAME.ERR
7B97-	D0	0A	8095	BNE SEND.ERR
7B99-	A9	33	8100 L.ERR	LDA #LOAD.ERR
7B9B-	D0	06	8105	BNE SEND.ERR
7B9D-	A9	34	8110 S.ERR	LDA #SAVE.ERR
7B9F-	D0	02	8115	BNE SEND.ERR
7BA1-	A9	35	8120 T.ERR	LDA #TEXT.ERR
7BA3-	38		8125 SEND.ERR	SEC
7BA4-	20	71 81	8130	JSR ERMSG
7BA7-	20	4D 83	8135	JSR CRLF
7BAA-	60		8140	RTS
			8145 ;	
			8150 DEV.ERR	.DE 49

```

      8155 FNAME.ERR      .DE 50
      8160 LOAD.ERR       .DE 51
      8165 SAVE.ERR       .DE 52
      8170 TEXT.ERR       .DE 53
      8175 DCMD.ERR       .DE 54
      8180 ;
      8185 ;
      8190 ;
      8195 ;
      8200 DISK.CMD5      JSR ACCESS
      8205                JSR GET.FNAME
      8210                BCS F.ERR
      8215                LDA FNAME
      8220                CMP #'#
      8225                BEQ SET.DEV
      8230                CMP #'!
      8235                BEQ D.STATUS
      8240                CMP #'?
      8245                BEQ D.DIRECT
      8250                CMP #'/'
      8255                BEQ SEND.CMD
      8260 TO.DC.ERR      BNE DC.ERR      ;DISK COMMAND ERROR
      8265 ;
      8270 SET.DEV        LDA FNAME+1
      8275                JSR ASCNIB      ;CONVERT TO HEX
      8280                BCS D.ERR      ;NON HEX NUMBER
      8285                CMP #8
      8290                BCC D.ERR
      8295                CLC
      8300                STA DISK.DEV
      8305                RTS
      8310 ;
      8315 D.STATUS       LDA DISK.DEV ;CHECK DEVICE #
      8320                CMP #8
      8325                BCC D.ERR      ;...<8, ERROR
      8330                CLC
      8335                STA C.DEV
      8340                JMP DISKSTV
      8345 ;
      8350 D.DIRECT       LDA DISK.DEV ;CHECK DEVICE #
      8355                CMP #8
      8360                BCC D.ERR      ;...<8, ERROR
      8365                CLC
      8370                LDX #1
      8375                JMP DIRENTV
      8380 ;
      8385 SEND.CMD       LDA DISK.DEV
      8390                CMP #8
      8395                BCC D.ERR
      8400                STA C.DEV
      8405                LDY #0
      8410 S.CMD          LDA FNAME+1.Y      ;MOVE COMMAND UP ONE
      8415                STA FNAME.Y
      8420                INY
      8425                CPY FN.LEN      ;#CHRS IN NAME
      8430                BCC S.CMD
      8435                DEC FN.LEN
      8440                JSR DISKCMDV

```

```

7C12- B0 B2      8445      BCS TO.DC.ERR
7C14- 60          8450      RTS
                   8455 ;
                   8460 ;
                   8465 ;
                   8470 ;
                   =====
                   GET FILE NAME
                   =====
7C15- E6 D3      8475 GET.FNAME INC *TXTPTR ;MOVE PTR TO NEXT CHR
7C17- E6 D3      8480      INC *TXTPTR ;PAST COMMAND
7C19- 20 CC 00    8485      JSR CHRGET
7C1C- C9 22      8490      CMP #' " ;LABEL NAME?
7C1E- D0 11      8495      BNE F.ERR.RTS
7C20- A2 00      8500      LDX #0
7C22- 20 CC 00    8505 G.FNAME JSR CHRGET
7C25- C9 22      8510      CMP #' "
7C27- F0 0A      8515      BEQ END.FNAME
7C29- 9D 35 01    8520      STA FNAME,X
7C2C- E8         8525      INX
7C2D- E0 1E      8530      CPX #30 ;MAX 30 CHARS IN FNAME
7C2F- 90 F1      8535      BCC G.FNAME
7C31- 38         8540 F.ERR.RTS SEC
7C32- 60         8545      RTS
                   8550 ;
7C33- E0 00      8555 END.FNAME CPX #0 ;EMPTY FNAME?
7C35- F0 FA      8560      BEQ F.ERR.RTS ;YES, ERROR
7C37- 8E 54 01    8565      STX FN.LEN ;SAVE NAME LENGTH
7C3A- 20 CC 00    8570      JSR CHRGET ;PREVENT SN ERROR
7C3D- 18         8575      CLC
7C3E- 60         8580      RTS
                   8585 ;
                   8590 ;
7C3F- 23 4C 50    8595 CMD.TABLE .BY '#LP'
7C42- 15 7B      8600      .SI LOAD.PRG-1
7C44- 23 53 50    8605      .BY '#SP'
7C47- 60 7B      8610      .SI SAVE.PRG-1
7C49- 23 44 43    8615      .BY '#DC'
7C4C- AA 7B      8620      .SI DISK.CMDS-1
                   8625 ;
7C4E- 47 43 30    8630 BAS.START .BY 'GC000' $0D
7C51- 30 30 0D    8635      .BY '28672' $0D
7C54- 32 38 36    8640      .BY '80' $0D
7C57- 37 32 0D    8645      .BY '1?X=USR(&"A603",0)' $0D
7C5A- 38 30 0D    8650      .BY 'RUN' $0D
7C5D- 31 3F 58    8655      .BY 0
7C60- 3D 55 53    8660 ;
7C63- 52 28 26    8665 .EN
7C66- 22 41 36
7C69- 30 33 22
7C6C- 2C 30 29
7C6F- 0D
7C70- 52 55 4E    8650      .BY 'RUN' $0D
7C73- 0D
7C74- 00          8655      .BY 0
                   8660 ;
                   8665

```



```

0005 ; CROSS REFERENCED LABEL LISTING FOR SYM1541 AND PRIM1541
0010 ;
0015 / = EXTERNAL # = LINE DEFINED
0020
0025 SYMBOL ;VALUE CROSS-REFERENCES
0030 /ACC.VEC ;#A613 #0245 0570 0580
0035 /ACCESS ;#8B86 #0200 2845
0040 /APPFLG ;#A648 #0285 ****
0045 /ASCNIB ;#8275 #0140 2500
0050 /BAS.COLDV ;#A603 #0220 0545 0715 0720
0055 /BIT.CNT ;#00F9 #0070 3485 3575 4140 4205 4250
0060 ; 4265 4330
0065 /C.DEV ;#A64E #0310 1040 1410 1620 1710 1985
0070 ; 2925 2985
0075 /CMD.CHR ;#00FC #0080 0780 0800 0835 0870
0080 /CRLF ;#834D #0170 1765 2040 2245 2275 2555
0085 /CURAD ;#00FE #0090 1155 1180 1265 1270 1280
0090 ; 1520 1530 1560 1720 1730 1770
0095 ; 1875
0100 /DF.CHR ;#0153 #0105 3280 3520 3750 3830 3950
0105 /DF.FLG ;#A62E #0265 3240 3265 3910 3925
0110 /DISK.CHR ;#00F8 #0065 4300 4360
0115 /DISK.DEV ;#A647 #0280 0530 1035 1405 1615 1705
0120 ; 1980 2520
0125 /EOI.FLG ;#00FD #0085 3255 3270 3410
0130 /ERMSG ;#8171 #0130 ****
0135 /EXECUTE ;#8855 #0175 ****
0140 /FN.LEN ;#0154 #0110 1740 1785 1895 1930 2005
0145 ; 2060 2095 2625 2775 2800
0150 /FNAME ;#0135 #0100 1715 1725 1995 2070 2080
0155 ; 2320 2335 2340 2360 2385 2395
0160 ; 2435 2585 2785
0165 /INCCMP ;#82B2 #0155 1570
0170 /INCHR ;#8A1B #0180 2205 2495 2580
0175 /INCP3 ;#8293 #0145 1400
0180 /INSTAT ;#8386 #0165 2175
0185 /INTCHR ;#8A58 #0190 ****
0190 /INVEC ;#A660 #0330 ****
0195 /J0.VEC ;#A620 #0250 0695 0700
0200 /J4.VEC ;#A628 #0255 0675 0680
0205 /J5.VEC ;#A62A #0260 0635 0640
0210 /LASTCMD ;#A657 #0325 0785
0215 /NEW.DCV ;#A610 #0240 0565
0220 /NEW.INV ;#A60C #0235 0560
0225 /NEW.OUTV ;#A609 #0230 0555
0230 /NEW.URSV ;#A606 #0225 0550 0595 0600 0955
0235 /OUTCHR ;#8A47 #0185 0740 1775 2170 2410 2445
0240 ; 2565
0245 /OUTVEC ;#A663 #0335 ****
0250 /P1 ;#A64E #0305 0310
0255 /P2 ;#A64C #0300 ****
0260 /P2SCR ;#829C #0150 1195 1455
0265 /P3 ;#A64A #0295 1030 1185
0270 /PARNR ;#A649 #0290 0985 1380
0275 /PSHOVE ;#8208 #0135 0980
0280 /RAE.SETUPV ;#A600 #0215 0540 0655 0660
0285 /RAE.WARM ;#B003 0665 0670
0290 /SA.CMD ;#A62F #0270 ****

```

0295	/SCRA	;\$A63A	#0275	****				
0300	/SPACE	;\$8342	#0160	2115	2120			
0305	/STATUS	;\$00FC	#0075	0080	1160	1210	1215	1245
0310				1285	1295	1320	1855	1905
0315				2085	2150	2285	2675	2695
0320				2945	3005	4345	4745	4750
0325	/TECHO	;\$A653	#0315	2185	2200	2225		
0330	/TOUT	;\$8AA0	#0195	****				
0335	/TOUTFL	;\$A654	#0320	****				
0340	/URSVEC	;\$A669	#0340	0615	0620			
0345	/USRENT	;\$8035	#0125	****				
0350	/VIA1DDRA	;\$A003	#0355	2855				
0355	/VIA1DRA	;\$A00F	#0380	1220	1545	2865	3315	3325
0360				3490	3495	3555	3570	3785
0365				4000	4010	4270	4275	4305
0370				4405	4415	4450	4460	4495
0375				4540	4550	4585	4590	4695
0380	/VIA1IFR	;\$A00D	#0375	3600	4175	4645		
0385	/VIA1PCR	;\$A00C	#0370	****				
0390	/VIA1T2H	;\$A009	#0365	2880	3595	4170	4640	
0395	/VIA1T2L	;\$A008	#0360	2875				
0400	ACPTR	;\$74BA	#4130	0495				
0405	ACPTRV	;\$7033	#0495	1150	1175	1235	1870	2065
0410				2075	2125	2145		
0415	ADDBAK	;\$72ED	#2380	2355				
0420	ATN.CLR	;\$7461	#3785	3850	4060			
0425	ATN.ERR	;\$716B	#1350	1230	1555			
0430	ATN.LOW	;\$73D1	#3315	3300				
0435	BAS.COLD	;\$7A37	0705	0710				
0440	BAS.INIT	;\$7A27	0685	0690				
0445	BIT.ON	;\$7423	#3540	3525				
0450	BUS.DIR	;\$73D9	#3335	3755	3835			
0455	C.IDL	;\$7561	#4695	4705				
0460	CI1	;\$748C	#3935	3915				
0465	CI2	;\$7491	#3950	3930				
0470	CIOUT	;\$7481	#3910	0480				
0475	CIOUTV	;\$702A	#0480	1525	1535	1565	2790	
0480	CLK.OFF	;\$752E	#4450	3340	3470	3995		
0485	CLK.ON	;\$7525	#4405	3305	3405	3545	3855	4095
0490				4145				
0495	CONTROL	;\$73B1	#3235	4055				
0500	CRLF.OUT	;\$72B4	#2245	2165				
0505	CTROLD	;\$7323	#2555	2600				
0510	DCHR.OUT	;\$7295	#2170	2140				
0515	DEV.NP	;\$744F	#3665	3400				
0520	DIR	;\$7236	#1970	0925				
0525	DIR.CHK	;\$70E1	#0915	0900				
0530	DIR.ENTRY	;\$723D	#1985	0425				
0535	DIR.ERR	;\$7238	#1975	****				
0540	DIR.ERR2	;\$730F	#2470	2270	2505	2515		
0545	DIR.FIX	;\$7281	#2125	2135				
0550	DIR.LINE	;\$725F	#2060	2105	2255			
0555	DIR.NAME	;\$728A	#2145	2180	2235			
0560	DIRENTV	;\$700F	#0425	****				
0565	DISK.CLOSE	;\$7344	#2675	0435				
0570	DISK.CMD	;\$71C9	#1630	0415				
0575	DISK.OPEN	;\$7356	#2755	0440				
0580	DISK.ST	;\$71E2	#1715	0420				



0585	DISKCLOSEV	;\$7015	#0435	1130	1500	1650	2300	
0590	DISKCMDV	;\$7009	#0415	****				
0595	DISKOPENV	;\$7018	#0440	1095	1465	1635	2015	
0600	DISKSTV	;\$700C	#0420	****				
0605	DLY.1MS	;\$7553	#4635	3350				
0610	DLY1	;\$7558	#4645	4655				
0615	END.MSG	;\$70AD	#0755	0735				
0620	EOI.CHK	;\$72BB	#2265	2090	2155			
0625	ERR.RTN	;\$7167	#1320	1100	1170	1470	1925	2715
0630	ERR.RTN2	;\$7353	#2715	2680	2700	2770		
0635	EXT.CMDS	;\$70EF	#0955	0810	0845	0865	0940	
0640	GET.DEF	;\$73C2	#3275	3245				
0645	GET.NAME	;\$732B	#2575	0430	2615			
0650	GETNAMEV	;\$7012	#0430	1045	1395	1625		
0655	GN.CHR	;\$732D	#2580	2620				
0660	GN.RTS	;\$7340	#2625	2590				
0665	HEX2AD	;\$72CE	#2310	2110				
0670	INIT.MSG	;\$70A2	#0730	0750				
0675	L.ERROR	;\$70FE	#1010	****				
0680	L2	;\$70F2	#0980	0815				
0685	L2/0PARM	;\$7105	#1035	0990				
0690	L2/1PARM	;\$7102	#1030	1000				
0695	L2/CHK	;\$70B3	#0790	****				
0700	LISTEN	;\$73AC	#3200	0465				
0705	LISTENV	;\$7021	#0465	2990				
0710	LOAD	;\$710E	#1085	0405				
0715	LOAD.BYTE	;\$7140	#1205	1255	1290			
0720	LOADV	;\$7003	#0405	****				
0725	MESSAGE	;\$7570	#4770	0730				
0730	NEW.CMDS	;\$70AE	#0780	0605	0610			
0735	NO.DATA	;\$7370	#2810	2780				
0740	NU	;\$7312	#2495	0945				
0745	NU.CHK	;\$70E8	#0935	0920				
0750	NXTDIG	;\$72D0	#2315	2430				
0755	RAE.INIT	;\$7800	0625	0630				
0760	RAE.SETUP	;\$7810	0645	0650				
0765	RC.CHR	;\$7214	#1870	1910				
0770	RCV.BIT	;\$74F5	#4260	4200				
0775	RCV.CLK	;\$74C2	#4150	4155				
0780	RCV.END	;\$722E	#1930	1885	1900	1920		
0785	RCV.EOI	;\$74E6	#4230	4210				
0790	RCV.ERR	;\$722B	#1925	1865				
0795	RCV.LINE	;\$7208	#1845	1750				
0800	RCV.PGM	;\$7128	#1150	1115				
0805	RCV.T256	;\$74CA	#4165	4255				
0810	RCV.TM1	;\$74CF	#4175	4195				
0815	RCV.TM2	;\$74DD	#4205	4185				
0820	RCV.TM3	;\$74F9	#4270	4280	4290	4335		
0825	RCV.TM4	;\$7507	#4305	4315	4325			
0830	S.CHK.IDL	;\$7560	#4690	3205				
0835	S.ERROR	;\$7169	#1325	1390	1610	1640	1655	1700
0840			2470					
0845	S.OUT1	;\$73F4	#3425	3435				
0850	S.OUT2	;\$73FA	#3440	3450				
0855	S.OUT3	;\$7400	#3455	3415	3465			
0860	S.OUT4	;\$740D	#3490	3500	3580			
0865	S.WAIT	;\$7426	#3545	3535				
0870	S2	;\$7170	#1380	0850				

0875	S2/CHK	;\$70C0	#0825	0795				
0880	SAVE	;\$7183	#1450	0410				
0885	SAVE.BYTE	;\$71AA	#1545	1575				
0890	SAVEV	;\$7006	#0410	****				
0895	SC	;\$71BC	#1605	0885				
0900	SC.CHK	;\$70CD	#0860	0430				
0905	SECOND	;\$745B	#3750	0470				
0910	SECONDV	;\$7024	#0470	3000				
0915	SEND.NAME	;\$7164	#2785	2805				
0920	SEND.OUT	;\$73E3	#3380	3260	3940			
0925	SEND.PGM	;\$71A0	#1520	1485				
0930	SET.LSTN	;\$739B	#2980	1480	2685	2765		
0935	SET.ST	;\$756A	#4745	1355	3705	4245		
0940	SET.TALK	;\$738D	#2920	1110	1860	2030		
0945	SETUP.VIA	;\$7375	#2840	0445				
0950	SETUPVIAV	;\$701B	#0445	0520	1085	1450	1845	
0955	SRL.0	;\$7540	#4540	3530	3845	4230	4340	
0960	SRL.1	;\$7537	#4495	3290	3345	3385	3540	4100
0965			4160					
0970	SRL.IC	;\$7549	#4585	3390	3425	3440	3455	3615
0975			3860	4150	4190	4595		
0980	ST	;\$71D8	#1695	0905				
0985	ST.CHK	;\$70DA	#0895	0880				
0990	ST.ERR	;\$71DA	#1700	1975				
0995	ST.LP	;\$71FB	#1770	1790				
1000	ST.RTS	;\$7207	#1800	1755	2020			
1005	START	;\$7036	#0520	0395	0575			
1010	SUBEM	;\$72D2	#2320	2375				
1015	SUBTBL	;\$730B	#2455	2330	2350	2390		
1020	SYM.END	;\$7600	#4820	****				
1025	T.OUT	;\$7454	#3705	4220				
1030	T.SA	;\$747A	#3860	3865				
1035	TALK	;\$73A9	#3165	0460				
1040	TALK.SA	;\$746A	#3830	0475				
1045	TALKSAV	;\$7027	#0475	2940				
1050	TALKV	;\$701E	#0460	2930				
1055	TIME.OUT	;\$7452	#3700	3515	3610			
1060	UNLISTEN	;\$74A5	#4050	0490				
1065	UNLISTENV	;\$7030	#0490	1490	2690	2815		
1070	UNLST1	;\$74AA	#4060	3720				
1075	UNLST2	;\$74AD	#4070	4235	4355			
1080	UNLST3	;\$74B0	#4080	4085				
1085	UNTALK	;\$7496	#3990	0485				
1090	UNTALKV	;\$702D	#0485	1120	1935	2290		
1095	VEC.SETUP	;\$703E	#0535	****				
1100	WAIT.ACK	;\$7440	#3600	3625				
1105	WAIT.SPC	;\$72A6	#2205	2215				

```

0005 ; CROSS REFERENCED LABEL LISTING FOR RAE1541 AND BAS1541
0010 ;
0015           / = EXTERNAL           # = LINE DEFINED
0020
0025 SYMBOL      ; VALUE              CROSS-REFERENCES
0030 /ACC         ; #00ED             7455      7495
0035 /ACCESS     ; #8B86             5255      6035      6100      7125      7215      7360
0040            ;                   7790      7955      8200
0045 /APPFLG     ; #A648             7795      7835
0050 /ASCNIB     ; #8275             6565      8275
0055 /BASWARM    ; #C27E             #7090      7315
0060 /BUFADR     ; #6F00             #5230      5390      5400
0065 /BUFF       ; #001E             #7030      7660
0070 /C.DEV      ; #A64E             5685      6305      6480      6620      6670      7815
0075            ;                   7975      8335      8400
0080 /CHRGET     ; #00CC             #7050      7750      8485      8505      8570
0085 /CHRGOT     ; #00D2             #7055      7580
0090 /CRLF       ; #834D             8135      ****
0095 /CRT.BUF    ; #0135             #5145      6040      6050      6395      6560
0100 /CUR.SAVE   ; #011C             #5115      5880      5890
0105 /CURAD     ; #00FE             5725      5735      5755      5885      5895      5910
0110            ;                   5935      5955      6045      6055      6060      6105
0115            ;                   6135      6185      7890      7895
0120 /DCMD.ERR   ; #0036             #8175      8080
0125 /DEV.ERR    ; #0031             #8150      8070
0130 /DISK.DEV   ; #A647             5520      5680      6225      6300      6460      6575
0135            ;                   6600      6650      7800      7960      8300      8315
0140            ;                   8350      8385
0145 /DISKC.VEC  ; #00EC             #5050      5345      5355
0150 /DISKI      ; #00EE             #5055      5310      7070
0155 /DISKI.VEC  ; #00F6             #5070      5325      5335
0160 /DISKO      ; #00EF             #5060      5315      7075
0165 /DISKO.VEC  ; #00F4             #5065      5375      5385
0170 /ERMSG      ; #8171             5795      8130
0175 /ERROR      ; #800E             #5165      5805      5820      5995      6005
0180 /EXECUTE     ; #8855             5280      7150
0185 /FILE.NO     ; #0110             #5105      5545      5615
0190 /FIRST       ; #0108             #5100      5870
0195 /FN.LEN      ; #0154             6195      8425      8435      8565
0200 /FNAME       ; #0135             6165      8215      8270      8410      8415      8520
0205 /FNAME.ERR   ; #0032             #8155      8090
0210 /HEND        ; #012B             #5140      5630      5635      6330      6340
0215 /HEX/DEC     ; #0111             #5110      5535      5660
0220 /HFILE/NO    ; #0128             #5130      5610
0225 /HIMEM       ; #0087             #7045      7910      7920
0230 /HSTART      ; #0129             #5135      5620      5625      6310      6320
0235 /INVEC       ; #A660             7220      7225      7250      7255      7390      7400
0240 /LBLHADR     ; #6EFC             #5225      5470      5480
0245 /LBLLADR     ; #6000             #5220      5450      5460
0250 /LINE.ADR    ; #B214             #5170      5875
0255 /LOAD.ERR    ; #0033             #8160      8100
0260 /NEW.DCV     ; #A610             5365      6440
0265 /NEW.INV     ; #A60C             7175      7180      7230      7235      7340      7385
0270            ;                   7395
0275 /NEW.OUTV    ; #A609             7285      7290      7365      7375      7430      7600
0280 /OUTVEC      ; #A663             7275      7280      7305      7310      7370      7380
0285 /P2          ; #A64C             5700      5710      6315      6325      7870      7875
0290            ;                   8000      8005

```

0295	/P3	;\$A64A	5265	5275	5690	6335	6345	7135
0300		;	7145	7820	8020	8025		
0305	/PSAD	;\$007B	#7035	7860	7865	7990	7995	
0310	/PURECL	;\$00C8	#5040	5395	5405			
0315	/RAE.WARM	;\$B003	#5160	5495				
0320	/RUN.CMD	;\$C707	#7100	7945				
0325	/SAVE.ERR	;\$0034	#8165	8110				
0330	/SCRA	;\$A63A	7165	7170	7195	7200		
0335	/SCRATCH	;\$C458	#7095	7160				
0340	/SET.EOT	;\$E597	#5185	5775				
0345	/STEN	;\$0106	#5095	5475	5485			
0350	/STST	;\$0104	#5090	5455	5465			
0355	/TAPE.GET	;\$EF68	#5190	5550				
0360	/TAPE.PUT	;\$EF95	#5195	6235				
0365	/TAPE0.OFF	;\$E30F	#5180	6255				
0370	/TAPE1.OFF	;\$E318	#5175	5570				
0375	/TECHO	;\$A653	7270	7410				
0380	/TEND	;\$0126	#5125	5730	5740			
0385	/TEXT.ERR	;\$0035	#8170	8120				
0390	/TPRES	;\$00D3	#5045	5860	7060			
0395	/TSTART	;\$0124	#5120	5575	5695	5705	6260	
0400	/TXEN	;\$0102	#5085	5435	5445	5745	5760	
0405	/TXST	;\$0100	#5080	5415	5425			
0410	/TXTHADR	;\$5FFC	#5215	5430	5440			
0415	/TXTLADR	;\$0200	#5210	5410	5420			
0420	/TXTPTR	;\$00D3	#7060	8475	8480			
0425	/USRENT	;\$8035	7415	***				
0430	/VSAD	;\$007D	#7040	7845	7850	7900	7905	8010
0435		;	8015					
0440	/XREG	;\$00EE	#7070	7460	7500			
0445	/YREG	;\$00EF	#7075	7465	7505			
0450	APPEND.PRQ	;\$7B13	#7775	***				
0455	BAS.COLD	;\$7A37	#7160	***				
0460	BAS.INIT	;\$7A27	#7125	***				
0465	BAS.START	;\$7C4E	#8630	7130	7140			
0470	C.CHK	;\$7AF3	#7660	7690				
0475	CHK.GTYPE	;\$7899	#5660	5585				
0480	CMD.CHK	;\$7AF1	#7655	7745				
0485	CMD.TABLE	;\$7C3F	#8595	7665	7695	7705		
0490	CTROLC	;\$7A8F	#7350	***				
0495	D.DIRECT	;\$7BE7	#8350	8245				
0500	D.ERR	;\$7B8D	#8070	7810	7970	8280	8290	8325
0505		;	8360	8395				
0510	D.STATUS	;\$7BD9	#8315	8235				
0515	DC.ERR	;\$7B91	#8080	8260				
0520	DIR.LIST	;\$7A07	#6645	6425				
0525	DIRENTV	;\$700F	6675	8375				
0530	DISK.CD	;\$79CD	#6460	6435				
0535	DISK.CMDS	;\$7BAB	#8200	8620				
0540	DISK.CT	;\$78EA	#5850	5665				
0545	DISK.GET	;\$7876	#5560	5525				
0550	DISK.GET1	;\$78A3	#5680	5980				
0555	DISK.PUT	;\$7979	#6245	6230				
0560	DISK.RAE	;\$79B3	#6385	5340	5350			
0565	DISKMDV	;\$7009	6495	8440				
0570	DISKSTV	;\$700C	6625	8340				
0575	END.FNAME	;\$7C33	#8555	8515				
0580	ERR31	;\$78E2	#5815	5765				



0585	ERR32	;	#7927	#6000	5830	6250	6535		
0590	ERR33	;	#78DA	#5795	5720	6530			
0595	ERR34	;	#7922	#5990	5915	6085	6180	6525	
0600	F.ERR	;	#7B95	#8090	7830	7985	8210		
0605	F.ERR.RTS	;	#7C31	#8540	8495	8560			
0610	FND.CT	;	#7904	#5905	5925	5945	5965		
0615	G.FNAME	;	#7C22	#8505	8535				
0620	G.NAM	;	#792E	#6035	6490				
0625	GET.FNAME	;	#7C15	#8475	7825	7980	8205		
0630	GET.NAM	;	#792C	#6030	5675	6295			
0635	GET.VEC	;	#7866	#5520	5320	5330			
0640	GN.DELIM	;	#7939	#6060	6090				
0645	GN.END	;	#796D	#6195	6150	6160			
0650	GN.MOV1	;	#794B	#6105	6125				
0655	GN.MOVE	;	#7946	#6095	5975				
0660	GNM.LP	;	#7954	#6130	6070	6190			
0665	GNM.LP1	;	#7955	#6135	6115				
0670	INPUT	;	#7A8A	#7340	7185	7190	7240	7245	
0675	L.ERR	;	#7B99	#8100	7885				
0680	L.PRGI	;	#7B3B	#7860	7840				
0685	L.PRGI2	;	#7B3F	#7870	7855				
0690	L.PRGI3	;	#7B58	#7925	7915				
0695	LOAD.PRGI	;	#7B16	#7785	7940	8600			
0700	LOADV	;	#7003	5715	7880				
0705	LOCAL.DEV	;	#79EC	#6555	6405				
0710	NEWCMDS	;	#7AE9	#7625	7590				
0715	NEXT.CMD	;	#7B09	#7725	7670				
0720	NXT.ERR	;	#7B10	#7750	****				
0725	OUTPUT	;	#7ACA	#7535	7295	7300			
0730	PRT.ST	;	#79F9	#6600	6415				
0735	PULL2	;	#7ACF	#7545	7555				
0740	PULL3	;	#7AEB	#7630	7640				
0745	PUT.DATA	;	#798A	#6290	6270				
0750	PUT.VEC	;	#7971	#6225	5370	5380			
0755	RAE.COLD	;	#7A17	#6685	5260	5270			
0760	RAE.INIT	;	#7800	#5255	****				
0765	RAE.SETUP	;	#7810	#5305	****				
0770	RESTORREG	;	#7AC3	#7495	7560	7605			
0775	RESTORXY	;	#7AC5	#7500	****				
0780	RUN.PRGI	;	#7B5B	#7940	****				
0785	S.CMD	;	#7C00	#8410	8430				
0790	S.ERR	;	#7B9D	#8110	8035				
0795	SAVE.PRGI	;	#7B61	#7955	8610				
0800	SAVEREG	;	#7ABC	#7455	7535				
0805	SAVEV	;	#7006	6350	8030				
0810	SAVEXY	;	#7ABE	#7460	****				
0815	SEND	;	#7AB9	#7430	7355				
0820	SEND.CMD	;	#7BF4	#8385	8255				
0825	SEND.ERR	;	#7BA3	#8125	8075	8085	8095	8105	8115
0830	SEND2	;	#7AE3	#7600	7575				
0835	SET.DEV	;	#7BC8	#8270	8225				
0840	T.ERR	;	#7BA1	#8120	7925				
0845	TO.DC.ERR	;	#7BC6	#8260	8445				
0850	TO.ERR32	;	#78E7	#5830	5565				
0855	TO.ERR33	;	#79E6	#6530	6355	6500			
0860	TO.ERR34	;	#79E3	#6525	6390				
0865	TOO.ERR32	;	#79E9	#6535	6470	6570	6610	6660	
0870	WEDGE	;	#7A53	#7215	7420				
0875	WEDGE2	;	#7A6C	#7265	7205				

```

0010 ;          SYM DOS UTILITY USER NOTE #1
0020 ;
0030 ;          BY RONALD A. JORDAN
0040 ;          NOVEMBER 19, 1984
0050 ;
0060 ;
0070 ; ENCLOSED IS YOUR LONG AWAITED FIRST UTILITY DISK.
0080 ; ALL PROGRAMS FUNCTION AS DESCRIBED, HOWEVER THEY
0090 ; HAVE NOT BEEN EXTENSIVELY TESTED AND MINOR PROBLEMS
0100 ; ARE POSSIBLE. I HAVE NOT HAD ANY PROBLEMS WITH
0110 ; THE PROGRAMS. IF YOU FIND A PROBLEM DROP ME A NOTE
0120 ; SO I CAN MAKE THE NECESSARY CHANGES.
0130 ;
0140 ; I AM CONTINUING TO WORK ON THE BASIC DOS ENHANCEMENTS.
0150 ; THE OPEN AND CLOSE COMMANDS STILL HAVE A FEW BUGS, WHICH
0160 ; I HOPE TO HAVE WORKED OUT BY CHRISTMAS. WHEN THIS GREATLY
0170 ; ENHANCED BASIC DOS INTERFACE IS COMPLETED, IT WILL BE OFFERED ON
0180 ; DISK FOR THE SAME AMOUNT AS THE UTILITY DISK ($15). ANY ADDITIONAL
0190 ; PROGRAMS AVAILABLE AT THAT TIME WILL ALSO BE INCLUDED.
0200 ;
0210 ; ADDR.RAE
0220 ; -----
0230 ; THIS PROGRAM EXAMINES THE PROGRAM LOAD ADDRESS.
0240 ; LOAD THE PROGRAM ADDR:200.OBJ AT $200 AND USE THE
0250 ; MONITOR COMMAND .G 200 TO START. BE SURE TO FIRST
0260 ; NOTE THE NAMES OF THE PROGRAMS OF INTEREST.
0270 ;          CTL D TO REENTER PROGRAM NAME
0280 ;          CTL X TO EXIT TO MONITOR
0290 ;
0300 ;
0310 ; TRKSECTOR.RAE
0320 ; -----
0330 ; THIS PROGRAM WILL READ AND WRITE ANY SECTOR ON THE DISK.
0340 ; LOAD THE PROGRAM TRKSEC:200.OBJ AT $200 AND START AS ABOVE.
0350 ; THE SECTOR OF INTEREST IS READ INTO MEMORY AT $1000. A SECTOR
0360 ; ON THE 1541 IS 256 BYTES. USE THE MONITOR .V AND .M COMMANDS
0370 ; TO EXAMINE AND MAKE CHANGES. THE TRACK AND SECTOR MUST BE
0380 ; ENTERED AS A DECIMAL NUMBER (IE 12 HEX = 18 DEC, ETC). THEY
0390 ; MAYBE ONE OR TWO DIGIT NUMBERS (IE 01 = 1).
0400 ;
0410 ;
0420 ; READDOS.RAE
0430 ; -----
0440 ; THIS PROGRAM READS THE 1541 DOS ROM'S INTO MEMORY. NO PROMPTS
0450 ; ARE PROVIDED. THE PROGRAM TAKES SEVERAL MINUTES TO DOWN LOAD
0460 ; THE ROMS TO RAM STARTING AT $1000 FOR 16K. THE PROGRAM CAN
0470 ; BE EXITED BY THE BREAK KEY.
0480 ;
0490 ;
0500 ; BACKUP.RAE
0510 ; -----
0520 ; THIS PROGRAM WILL MAKE A BACKUP OF 1541 DISK. THIS PROGRAM
0530 ; WORKS LIKE THE C-64 BASIC PROGRAM CALLED 1541 BACKUP, EXCEPT
0540 ; THIS PROGRAM IS ALL M-L AND RUNS FASTER. THE PROGRAM
0550 ; BACKUP:200.OBJ MUST BE LOADED AT $200 AND RUN AS ABOVE. BE
0560 ; SURE TO HAVE A BLANK FORMATTED DISK FOR YOUR DESTINATION
0570 ; DISK. THE PROGRAM USES A RAM BUFFER FROM $1000-$6FFF. THE LARGER
0580 ; THE BUFFER THE FEWER TIMES THE DISKS MUST BE SWITCHED. THERE
0590 ; ARE TWO METHODS OF BACK UP: 1. THE BAM OPTION AND 2. THE DIRECT
0600 ; OPTION. THE BAM METHOD LOOKS AT ONLY THE SPACE USED BY PROGRAMS
0610 ; ON THE DISK (BLOCK ALLOCATION MAP). THIS METHOD IS MUCH FASTER
0620 ; AND RECOMMENDED UNLESS THE DISK IS FULL. THE DIRECT METHOD
0630 ; COPIES THE ENTIRE DISK AND IS MUCH SLOWER AND REQUIRES SWITCHING
0640 ; THE DISKS SEVERAL TIMES.

```